5-16-2018

Semi-Supervised Learning for Diagnosing Faults in Electromechanical Systems

Ehsan Hallaji
University of Windsor

SEMI-SUPERVISED LEARNING FOR DIAGNOSING FAULTS IN ELECTROMECHANICAL SYSTEMS

by

Ehsan Hallaji

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2018

Semi-Supervised Learning for Diagnosing Faults in Electromechanical Systems

by

Ehsan Hallaji

APPROVED BY:

L. Rueda
School of Computer Science

R. Razavi-Far
Department of Electrical and Computer Engineering

M. Saif, Advisor
Department of Electrical and Computer Engineering

May 7, 2018

Declaration of Co-Authorship and Previous Publication

I. Co-Authorship

I hereby declare that this thesis incorporates material that is the outcome of my research under the supervision of Dr. Mehrdad Saif. This thesis also incorporates the results of my research undertaken in collaboration with Dr. Rueda in Chapter 3; Dr. Razavi-Far in Chapters 3-5; Dr. Kia, Dr. Heano and Dr. Capolino in Chapter 4; Dr. Ditzler in Chapter 5; and Ms. Farajzadeh-Zanjani in Chapters 3-4. In all cases, the key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by the author, and the contribution of co-authors was primarily through the provision of study materials. Dr. Rueda provided feedback on the refinement of dimensionality reduction procedure; Dr. Razavi-Far provided consultation on semi-supervised learning and fault diagnosis; Dr. Kia, Dr. Heano, Dr. Capolino provided data and information for the gearbox case; Dr. Ditzler provided feedback on refinement of COMPOSE implementation; Ms. Farajzadeh-Zanjani contributed to the signal analysis.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-authors to include the co-authored material in my thesis.

I certify that, with the above qualification, this thesis and the research to which it refers is the product of my own work.

II. Previous Publication

This thesis partly includes the original papers that have been previously submitted, to be submitted, or published in peer reviewed journals and conferences as provided in the following table.

Thesis Chapter	Publication title/full citation	Publication status
Chapters 1-6	R. Razavi-Far, E. Hallaji, M. Saif and L. Rueda, "A Hybrid Scheme for Fault Diagnosis with Partially Labeled Sets of Observations," 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 61-67.	published
Chapters 1-6	R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani and M. Saif, "A Semi-supervised Diagnostic Framework Based On the Surface Estimation of Faulty Distributions," 2017.	submitted
Chapters 1-6	R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, M. Saif, S. H. Kia, H. Heano and G. A. Capolino, "Semi-Supervised Deep-Learning Procedure for Diagnosing Simultaneous Gear Faults," 2018.	to be submitted
Chapters 1-6	R. Razavi-Far, E. Hallaji, M. Saif and G. Ditzler, "A Novelty Detector and Extreme Verification Latency Model for Nonstationary Environments," IEEE Transactions on Industrial Electronics, 2018, pp. 1-1	accepted for publication

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Safe and reliable operation of the systems relies on the use of online condition monitoring and diagnostic systems that aim to take immediate actions upon the occurrence of a fault. Machine learning techniques are widely used for designing data-driven diagnostic models. The training procedure of a data-driven model usually requires a large amount of labeled data, which may not be always practical. This problem can be untangled by resorting to semi-supervised learning approaches, which enables the decision making procedure using only a few numbers of labeled samples coupled with a large number of unlabeled samples. Thus, it is crucial to conduct a critical study on the use of semi-supervised learning for the purpose of fault diagnosis.

Another issue of concern is fault diagnosis in non-stationary environments, where data streams evolve over time, and as a result, model-based and most of the data-driven models are impractical. In this work, this has been addressed by means of an adaptive data-driven diagnostic model.

Acknowledgements

This work would not have been possible without support and help of many peoples. I would like to express my sincere gratitude to those who encouraged and supported me through this period of research and intensive learning.

Special mention goes to my enthusiastic supervisor Dr. Mehrdad Saif, Dean of Faculty of Engineering. My Master's has been an amazing experience and I thank him wholeheartedly, not only for his encouragement and tremendous academic support but also for giving me so many wonderful opportunities.

Similar, profound gratitude goes to Dr. Roozbeh Razavi-Far of the Department of Electrical and Computer Engineering, who has been a truly dedicated mentor. As the internal reader of this thesis, he was very motivative and always provided me with many insightful comments during the development of the ideas in this thesis.

I am also hugely appreciative of Dr. Luis Rueda of the School of Computer Science as the external reader of this thesis for his time and helpful comments, which improved this work to a large extent.

Last but most importantly, I must express my very profound gratitude to my beloved family, my parents and my sister, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

IM	Induction Motor
CWRU	Case Western Reserve University
IRF	Inner Race Fault
ORF	Outer Race Fault
BF	Ball Fault
SL	Supervised Learning
UL	Unsupervised Learning
SSL	Semi-supervised Learning
FFT	Fast Fourier Transformation
LMD	Local Mean Decomposition
FE	Feature Extraction
FS	Feature Selection
FR	Feature Reduction
DR	Dimensionality Reduction
EM	Expectation Maximization
WPT	Wavelet Packet Transform
EMD	Empirical Mode Decomposition
SSA	Singular Spectrum Analysis

SVD	Singular Value Decomposition
FDS	Fault Diagnosis System
SDA	Semi-supervised Discriminant Analysis
SSDR	Semi-supervised Dimensionality Reduction
FME	Flexible Manifold Embedding
TRCFS	Trace Ratio Criterion Feature Selection
SFSS	Structural Feature Selection with Sparsity
GLP	Graph-based Label Propagation
LDS	Low Density Separation
SSELM	Semi-supervised Extreme Learning Machine
LapSVM	Laplacian Support Vector Machine
ASSEMBLE	Adaptive Semi-supervised Ensemble
SemiBoost	Semi-supervised Boosting
RegBoost	Regularized Boosting
S3AL	Semi-supervised Smooth Alpha Layering
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
SSDLN	Semi-supervised Deep Ladder Network

SCSVIF Stator Current Space Vector Instantaneous Phase

SCSVIA Stator Current Space Vector Instantaneous Amplitude

IMF Intrinsic Mode Function

PF Product Function

EVL Extreme Versification Latency

EVLC Extreme Versification Latency Classifier

APT Arbitrary Sub-population Tracker

SCARGC Stream Classification Algorithm Guided by Clustering

GMM Gaussian mixture model

RMS Root Mean Square

E-CUSUM Extended Cumulative Sum

COMPOSE Compact Object Sample Extraction

DISCOVERY Drift and Novelty Class Detection and Adaptation under Extreme Vitrification Latency

Chapter 1

Introduction

Many situations naturally arise in the industry where resilient technologies and software algorithms are required to keep a system stable. A diagnostic system is usually required to maintain the system performance and minimize the downtime. As a result, the development and deployment of a robust Fault Diagnosis System (FDS) are of critical concern in applied industrial applications, and the computational intelligence community has begun to propose solutions to this problems [3].

1.1 Motivation

The challenge of providing a robust FDS can be addressed using model-based and data-driven techniques. Model-based approaches make use of predictive models that are based on the prior knowledge of different system states [4, 5]. These models are used to detect a fault and identify the type of fault that has occurred. However, obtaining these models is typically complex and difficult in practice. To overcome this issue, data-driven techniques have been widely used in the design of the FDS systems. These data-driven diagnostic models usually make use of intelligent techniques to train predictive models based on a set of samples [6]. These trained predictive models are then used to predict the system states in online applications.

These data-driven techniques generally follow the Supervised Learning (SL) paradigm that requires labeled data to be available from all known classes (i.e., normal operations and different types of faults) to build a model [7]. SL models typically work well for diagnosing faults [8]; however, their effectiveness, in practice, relies on the

amount of available labeled data. Unfortunately, acquiring a large volume of labeled data becomes infeasible because the labels need to come from a human expert, which is commonly an expensive operation (both in terms of time and money). Note that collecting data is typically easy and it is the process of obtaining the labels that is expensive. On the other hand, the field of Semi-Supervised Learning (SSL) focuses on techniques that leverage both labeled and unlabeled data to build a predictive model [9]. Exploiting the unlabeled data to extract information enables the algorithm to construct a model with much less required prior knowledge than SL [10].

1.2 Contributions

The aim of this work is to design hybrid fault diagnostic frameworks that work with a few number of labeled samples and a large portion of unlabeled samples. To evaluate each of the designed frameworks, various state-of-the-art approaches have been devised for different modules of the framework such as Feature Extraction (FE), Dimensionality Reduction (DR), and classification. This enables comparative studies in which advantages and disadvantages of each technique can be perceived for the respective application. These comparative studies are mainly focused on SSL, which is leveraged for DR and classification. The studies reported in this work are very informative and beneficial in the sense that SSL is seldom studies for the sake of fault diagnosis.

Moreover, novel semi-supervised classification algorithms have been proposed in this work that are utilized for fault classification in the designed diagnostic frameworks. Further contributions have been made in the field of real-time classification in non-stationary environments and addressing the occurrence of new classes in the data stream. Data-driven FDS approaches are usually trained based on collected data in stationary environments (i.e., data are sampled from an independent and identically distributed (i.i.d.) process). On the other hand, collecting representative data

is often a dynamic process of successive data acquisition campaigns. In such non-stationary environments, data patterns become available successively, over a period of time. Therefore, designing efficient data-driven schemes are typically preferred, since these techniques are more robust against arbitrary data distributions and types of faults [11]. Furthermore, they do not rely on seasonal changes in patterns (i.e., a stationarity assumption). Our setting assumes that data arrive in a non-stationary environment, where sensor data collection forms a data stream [12]. More details regarding the novelties in this work are included in the following subsection.

1.3 Novelties

In addition to the vast comparative studies on the application of various state-of-the-art SSL algorithms in FDS, novel algorithms are also proposed to maximize the overall performance of the designed framework:

1.3.1 Semi-Supervised Smooth Alpha Layering

A novel semi-supervised classifier, called Semi-Supervised Smooth Alpha Layering (S3AL) has be introduced in this work. S3AL aims to perform multiclass classification by resorting to an inductive learning procedure. In this algorithm, the structure of data is captured using α-Shape [13]. α-Shape is a strong tool, which is mostly utilized for the surface estimation. However, to our knowledge, it is rarely exploited for SSL and designing a FDS.

1.3.2 Drift and Novelty Class Detection and Adaptation under Extreme Verification Latency

Diagnosing faults in non-stationary environments is a challenging task. On the other hand, providing FDS with external updates is not feasible, since labeling the unlabeled

data is an expensive and offline task. To overcome these issues, recent works have been developed to address Extreme Verification Latency (EVL), which is when the labeled data is merely available at the first time step. A more realistic learning setting for the design of FDS aligns better with EVL. Although EVL learners can be used for diagnosing faults from the data stream, they have some restricting assumptions. EVL assumes that the labeled data is only provided at the initial time step, and the subsequent samples are fully unlabeled, which is a more realistic learning scenario for FDS. EVL classifiers (EVLC) are mostly limited to work with gradual drifts and the fixed number of classes, while there is a need to work with both abrupt and gradual changes in the data stream, and, moreover, handle new classes in the subsequent instalments, without any external updates.

One of the primary contributions of this work is to design a diagnostic framework based on any EVL classifier, where the restricting assumptions are addressed by resorting to a detection and adaption module. This work proposes a novel framework, which contains two main modules. These include a double-stage detector and a classification module. To work with both abrupt and gradual changes and handle new classes without any external updates, a double-stage detector has been devised in the proposed framework. This double-stage detector initially uses the Extended CUmulative SUM (E-CUSUM) technique to detect abrupt changes and, then, a novelty detector to determine the presence of new unseen classes.

1.3.3 Affinity-based COMPOSE

Another contribution of the work is to develop a new EVLC that is a variant of the COMPact Object Sample Extraction (COMPOSE) algorithm [14], which is named as Affinity-based COMPOSE. The main difference between the Affinity-based COMPOSE and other variations of COMPOSE is the sampling procedure in which a prospective sampling procedure has been devised in order to compromise between the accuracy and the runtime.

1.4 Outline

The remainder of this study is organized as follows:

Chapter 2 initially describes the main challenges in designing data-driven fault diagnostic systems. Then, two case studies used in this work, induction motor and gearbox, are introduced. Afterwards, it presents the literature review on semi-supervised learning algorithms and its different categories. Firstly, the nature of semi-supervised learning is clarified. A brief review is then performed on semi-supervised dimensionality reduction. Finally, some of the semi-supervised classifiers that are used as comparative tools are introduced.

Chapter 3 presents the two hybrid diagnostic frameworks that are designed to address the problem of fault diagnosis under limited supervision. The first framework enables a brief study on the semi-supervised dimensionality reduction, while the second framework has a more complex scheme and focuses on semi-supervised classification. In addition, a novel semi-supervised classifier, called Semi-Supervised Smooth Alpha Layering (S3AL), is proposed in this chapter, which is designed to maximizes the performance of the second diagnostic framework. The experimental results are also included in this chapter.

Chapter 4 introduces a hybrid diagnostic framework that is able to address the problem of fault diagnosis with the limited supervision in a high dimensional feature space by making use of a semi-supervised deep learning procedure. Then, a comparative study is performed on state-of-the-art deep learning algorithms and the results are analyzed afterwards.

Chapter 5 presents a dynamic diagnostic framework to address the problem of fault diagnosis with the limited supervision in non-stationary environments. This novel framework is called Drift and Novelty Class Detection and Adaptation under Extreme Vitrification Latency (DISCOVERY), which utilizes a double stage detector that is proposed in this work. Moreover, a novel classifier, called Affinity-based

COMPOSE is proposed and devised in the DISCOVERY in order to maximize the speed for decision making and the diagnostic accuracy. Affinity-based COMPOSE is compared and studied within the designed framework. Finally, the experimental results are presented and analyzed in this chapter.

Chapter 6 presents the conclusion of this thesis. It first states what is the aim of this study and the considered problems in this work. Then, accomplished works in each chapter are reviewed and concluded providing an overview..

Chapter 2

Data-driven Fault Diagnosis

Diagnosing a fault in a system can be achieved through a detection and classification phase. Many solutions have been proposed for the fault diagnosis problem; however, most use either: (a) an unsupervised learning (UL) scheme for fault detection, or (b) SL algorithm for classification, respectively. SL strategies have been widely used in design of the data-driven diagnostic systems due to the high diagnostic accuracy. However, the SL strategy to design diagnostic systems is usually infeasible and expensive due to the unavailability of labels during the training phase.

Another key factor to determine the accuracy of the data-driven diagnostic systems is the quality of the input features. Various methods can be used to extract representative features from the raw sensory measurements; however, this may result in creation of the large pool of features including redundant ones, which further complicates the training procedure of the diagnostic models. As a result, there is a need to reduce the dimension of the feature space to improve the diagnostic accuracy. This has been widely used in many research works [15]; however, this work aims to overcome this issue by resorting to a semi-supervised deep learning network.

The input signals to the FDS are usually in the form of data stream and require fast and immediate predictions. Although, SL strategies are quite beneficial in stationary environments, more efficient solutions are required due to the dynamical behavior of the systems. In non-stationary environments, the data become available incrementally over time. The underlying distribution of data collected in subsequent installments may change due to concept drift ($\mathcal{CD}$). $\mathcal{CD}$, in practice, is a result of change in operational conditions (e.g., load variation) or state of the system (e.g., new classes

of faults are presented). $\mathcal{CD}$ includes any alternation in the collected signal brought about by a change in fault diameters or the IM speed. The $\mathcal{CD}$ can be divided into two major categories: abrupt ($\mathcal{CD}_{\mathcal{A}}$) and gradual ($\mathcal{CD}_{\mathcal{G}}$). Therefore, FDS require a solution that is incrementally updated over time. However, incremental learning procedure becomes more complicated in the presence of the unlabeled data stream and by the appearance of abrupt changes or novel classes of faults.

In this work, the aforementioned issues are addressed by resorting to semi-supervised learning approach. To assess the practicality of the proposed data-driven diagnostic systems for real-life problems, electromechanical systems are considered for the sake of this study, which are introduced in the following section.

2.1 Electromechanical Systems

Two main case studies are considered for evaluating the performance of the proposed diagnostic frameworks.

2.1.1 Induction Motors

The undeniable importance of induction motors (IMs) in industry resulted in a large number of research works in various domains [16, 17]. One of the greatest concerns, is designing an efficient fault diagnosis system, in which failures and malfunctions would be identified in order to inform the control unit or operators to make preventive decisions before system breakdown [18, 19].

Among the various failures in induction motors, almost 41% of them are caused by bearing defects [20]. Such defects are followed by many consequences such as costly repairs, system breakdown, and in the worst case, workers injuries. Thus, finding a reliable and robust fault diagnosis system is of predominant concern, to ensure the reliable operations of the IMs.

The Case Western Reserve University (CWRU) is one of the most popular cases in

the field of bearing fault diagnosis [21]. The CWRU data represents vibration signals generated by different defects. These vibration signals are raw and contains noise and high-frequency components, which complicate the diagnostic process. Thus, a pre-processing phase is required to prepare informative sets of features for the upcoming module.

For this case study, bearing defects are categorized into four different types namely as normal, inner, ball, and outer race defects under different conditions such as different speeds (1730, 1750, 1772, $1797rpm$) and defect widths (0.007, 0.014, $0.021in$).

2.1.2 Gearbox

Gears are the main component of mechanical power transmission systems in various industrial applications including aerospace, marine, railway, automobile and wind turbine. Health assessment and preventive maintenance of the gears are crucial for reliable, safe, optimal operations and can reduce the maintenance cost of electromechanical systems efficiently. The vibration-based condition monitoring has been considered as the most common method of gear fault diagnosis, since any mechanical imperfection modifies the response of gear's mechanical structure to external excitation and, hence, produces faulty signatures in the vibration signal [22, 23]. However, alternative methods based on oil debris, acoustic noise and acoustic emission analysis have been used for gear condition monitoring in electromechanical systems [24, 25]. Recently, the gear fault detection based on the electrical signature analysis has been proposed, which is cost effective, since it needs minimum installation changes in the system and do not need the installation of any extra sensors, since in most of industrial applications the electrical measurement is already available [26, 27]. In this regard, both stator current space vector instantaneous amplitude (SCSVIA) and stator current space vector instantaneous phase (SCSVIF) have been utilized for parallel shaft gear tooth fault detection [28, 29, 30].

2.2 Literature Review

Here, the nature of SSL problem and its categories are explained. Moreover, some of the state-of-the art algorithms that are used in this study for the purpose of fault diagnosis are explained.

In order to extract useful knowledge from the unlabeled samples $U = \{x_1, x_2, ..., x_{n_u}\}$, SSL algorithms usually find relationships between samples and relate them to the information acquired by the labeled samples $L = \{(x_1, y_1), (x_2, y_2), ..., (x_{n_l}, y_{n_l})\}$ to some extent, where the label y belongs to a class in $\Omega = \{\vartheta_1, \vartheta_2, ..., \vartheta_{n_\Omega}\}$. Such relationships are mainly extracted based on three fundamental semi-supervised assumptions (SSA):

1. **Manifold Assumption:** data is representable on a low-dimensional manifold. Graph-based schemes are widely used under this assumption [31]

2. **Cluster Assumption:** data samples that are clustered together are assumed to belong to the same class, i.e., the decision boundary between the two classes should pass through the low density regions [32]

3. **Smoothness Assumption:** samples in dense regions of a distribution should share the same class [33]. SSL can be used for Feature Selection (FS), Dimensionality Reduction (DR) and classification.

2.2.1 Semi-Supervised Feature Reduction

Semi-supervised learners usually require a representative set of features in order to achieve a good classification performance. Otherwise, learning on biased or non-informative features can result in the performance drop. In general, feature reduction (FR) provides the learner with a smaller set of informative features. FR approaches can be divided in two major categories:

1) *Dimensionality reduction* approaches project the samples onto a lower-dimensional space [34, 35, 36];

2) *Feature selection* approaches rank and select the most informative features and filter out the redundant features [37, 38].

However, majority of these FR approaches work in a SL or an unsupervised learning framework. The process of FR for partially labeled sets can be further improved by resorting to a SSL framework, which makes use of both labeled and unlabeled samples. In this study, five FR algorithms are utilized, which are explained in the following:

Semi-Supervised Discriminant Analysis (SDA)

SDA [34] exploits unlabeled data U as well as labeled data L to gain knowledge about the geometric structure of data in addition to the class separability. In other words, U is used to capture the geometric structure of the data, while L is used to maximize the between-class separability in the new feature space.

Semi-Supervised Dimensionality Reduction (SSDR)

SSDR [35] reduces the dimensionality of data by exploiting the unlabeled data and pairwise constraints, in which pairs of observations within the same class (must-link constraint) and in different classes (cannot-link constraint) are determined. SSDR aims to find a projection matrix in a way that both the specified pairwise constraints and the structure of original data are retained.

Flexible Manifold Embedding (FME)

FME [36] integrates the smoothness on the data manifold and the label fitness, in order to achieve an optimal projection matrix. It gains knowledge about the labels fitness using L, and the manifold structure using both L and U. The former is used to define the label fitness, and the latter is used to apply manifold smoothness and

the flexibility penalty. The goal is to find the optimum prediction labels, a linear regression function for projecting new data and a regression residue, which models the mismatch between predicted labels and the regression function.

Trace Ratio Criterion Feature Selection (TRCFS)

TRCFS [37] finds the best features through a filter-based approach. Using this approach, the within-class and the between-class scatter matrices are formed on the soft labels estimated by the label propagation, and the best features are then selected w.r.t. a noise insensitive trace ratio criterion.

Structural Feature Selection with Sparsity (SFSS)

SFFS [38] integrates the $l_{2,1}$ regularized FS and manifold learning to attain a semi-supervised FS scheme. By this mean, U and L are both used for selecting features jointly, while the correlation between them is considered at the same time.

2.2.2 Semi-Supervised Classification

Semi-supervised classifiers can be categorized into graph-based and cluster-based algorithms as follows:

1) *Graph-based* algorithms aim to capture the intrinsic structure of the data by constructing a graph on the data samples. Successful graph-based algorithms include label propagation [39], graph cut algorithms [40], and Low Density Separation (LDS) [41]. Graph-based algorithms are usually transductive, where the algorithm observes all the available samples and, then, classifies them at once. Transductive learning is usually accurate, since it observes all the unlabeled samples prior to prediction. However, when a new sample arrives, the procedure should be executed again, since there is not any specific model available. More recent graph-based SSL approaches maximize the margin, while minimizing the inconsistency on the manifold structure

of the data. These approaches mainly include manifold regularization [31] such as LapSVM, LapRLS [31] and Semi-supervised Extreme Learning Machines (SSELM) [42].

2) *Cluster-based* algorithms form a decision boundary in the low density regions. This can be usually done by maximizing the classification margin, as in transductive SVM [43] and semi-supervised SVM [32]. Cluster-based approaches can also be used in the boosting frameworks for the sake of SSL [44, 45, 33]. Most of the cluster-based algorithms are inductive learners, that aim to construct a predictive model on the training data, prior to prediction.

In this study, various semi-supervised classifiers have been considered, where some of them are explained in the following:

Graph-based Label Propagation (GLP)

GLP initially constructs a graph structure on all data points, i.e., $U \cup L$. Various approaches can be used for attaining this graph structure. In this work, this has been accomplished by means of harmonic function introduced in [46]. Once the graph is constructed, labels start to propagate from the available labeled samples on the graph.

Low Density Separation (LDS)

LDS initially leverages a graph-based approach to reduce the dimensionality of the data, and, then, it aims to maximize a classification margin. Although various advanced methods are available for SSL, they are mostly designed for low-dimensional datasets and as a result, they do not yield to their best performance for extremely high-dimensional data. LDS is chosen as it makes use of an intrinsic dimensionality reduction phase.

Semi-Supervised Extreme Learning Machine (SSELM)

SSELM is an extension of the Extreme Learning Machine (ELM) [47]. It firstly, constructs a hidden layer for a single layer feedforward network by random generation of the feature mapping. To compute the optimal weights between the output and the hidden layer, SSELM reformulates the ELM loss function to apply for the SSL problem. Such a modification is performed w.r.t. manifold regularization [31], in which an affinity matrix is utilized to penalize the inconsistency between samples so that samples in high-density regions are expected to share the same class, i.e., known as smoothness assumption.

Laplacian Support Vector Machine (LapSVM)

Similar to support vector machines, LapSVM aims to maximize a classification margin. LapSVM also minimizes the inconsistency on the data manifold structure simultaneously. In other words, LapSVM adds an smoothness penalty term to the SVM objective function, by which label smoothness is controlled on the captured manifold.

Adaptive Semi-Supervised Ensemble (ASSEMBLE)

ASSEMBLE [44] utilizes the concept of pseudo-class to exploit unlabeled data for maximizing a margin. Initially a supervised model is formed based on the available labeled data, and then, a prediction is made on U. The predicted labeles are referred as pseudolabels. Using these pseudolabels, a pseudo-margin can be maximized in an iterative procedure, where at each iteration the mispatch between model and prediction is penalized, and then, the pseudolabels are updated for the next iteration.

Semi-Supervised Boost (SemiBoost)

SemiBoost [45] initially creates pseudolabels based on a similarity matrix, which is formed by means of a radial basis function. Then, through an iterative procedure,

the most confident samples are chosen w.r.t. a confidence that is calculated using pairwise similarities. These confident samples are used to form a classification model that is assigned with an ensemble weight at the end of the iteration. This ensemble weight is computed w.r.t. the mismatch between the predictions and the expected predictions (i.e., the computed confidence for a sample indicates how likely it is to be in a class).

Regularized Boost (RegBoost)

RegBoost [33] uses density-based clustering and a similarity matrix to initially generate pseudolabels. After sampling the most confident samples and constructing an initial classification model is constructed. Then, the ensemble is updated by assigning the model with a weight and including it in the current ensemble. This weight is calculated by penalizing the misclassification on labeled data applying the smoothness assumption on the unlabeled data. The latter is performed locally in the neighbourhood of each unlabeled sample.

2.3 Summary

This chapter initially explains the main challenges in designing data-driven fault diagnosis systems. Firstly the problem of fault diagnosis under limited supervision is stated, in which the assumption is that only a few numbers of labeled samples along with a large number of unlabeled samples are available. Then, a more complicated case is considered where the input space is very high-dimensional. Afterwards, the challenges of fault diagnosis in non-stationary environments with the presence of concept drift are explained. In addition, two cases of electromechanical systems, namely induction motors and gearbox, are introduced. Finally, a literature review on the SSL and the utilized algorithms is presented.

Chapter 3

Fault Diagnosis with Limited Supervision

Most of data-driven techniques utilize a Supervised Learning (SL) strategy, which requires a large set of labeled (i.e., faulty or normal states) samples that are collected a priori [48]. SL algorithms are usually accurate if provided with a representative set of samples. Obtaining these sets of samples is not a difficult task as the data is usually collected without the human interference by means of sensors. However, the process of labeling (i.e., whether they are faulty or normal) the collected samples, requires the human knowledge about the system states. For this reason, most of the FDS systems designed based on the SL strategy are not useful in real applications, where a large number of unlabeled samples are collected. It is very important then to build a model in a more efficient manner for designing a practical FDS.

To overcome this issue, Semi-Supervised Learning (SSL) strategies have been used in design of the FDS systems [49, 50] that facilicates the procedure of model construction with a partially labeled set of samples. Besides, only a few number of labaled samples are used to construct the model, and, thus, the process of labeling the unlabeled samples is no longer required. These SSL strategies usually make use of the labeled data in order to extract useful information from the unlabeled samples to compensate the lack of enough labeled samples. For this reason, various diagnostic systems are designed based on SSL strategies to improve the efficiency and the accuracy of the FDS [51, 52].

The aim of this chapter is to design hybrid fault diagnostic frameworks that work with a few number of labeled samples and a large portion of unlabeled samples. The designed frameworks consist of three main steps for feature extraction (FE) from

the input signals, feature reduction (FR) of the extracted sets, and decision making. Various state-of-the-art approaches are devised in each step. Comparative studies are then performed on all approaches in order to find the best combination of the approaches that leads to the best performance.

3.1 Problem Formulation

Here, the proposed frameworks are applied for diagnosing bearing defects in induction motors. This study is based on the standard case study from the Case Western Reserve University (CWRU) [21]. Bearing Data Center contains different vibration signals for the normal and faulty conditions. Since, in reality IM conditions may vary (e.g., the defect width or motor load may change), we have considered four different scenarios gathered from the CWRU data center. Table 3.1 shows detailed information for each scenario. Two different motor loads of 0 and 1 hp with two different defect widths of 0.007 inch and 0.014 inch are considered in this study. The data collection was performed at 12000 sample per second for drive end experiments. In addition, four different defect conditions including Normal, Inner Race Fault (IRF), Outer Race Fault (ORF) and Ball Fault (BF) are considered that result in a multiclass classification problem.

In this chapter, two diagnostic frameworks are designed. One enables a brief comparative study on semi-supervised dimensionality reduction, while the other one is the main diagnostic framework proposed in this chapter, and it is focused on semi-supervised classification. The latter is referred as the main experiment in this chapter. The experimental setting for each of them is explained in the following.

3.1.1 Study on Semi-Supervised Dimensionality Reduction

In order to prepare the initial data for the DR module, signal data from CWRU dataset is segmented into 320 samples, in which 80 samples are available for each

class of Normal, IRF, BF, and ORF. These are representative samples from the drive-end signals with the defect width of $0.007in$ and $0.021in$. In total, 112 features are extracted from the raw signals.

3.1.2 Main Experiment

The length of the vibration signal for each motor condition is 102400, which is segmented later according to a fixed window of size 1024 resulting in 100 segments. Each scenario contains representative samples of all classes (i.e., Normal, IRF, ORF and BF). First scenario contains samples of the lowest load and the smallest width for all defects. In the second and third scenarios, IM is only subjected to the change in the motor load and the defect widths, respectively. The last scenario contains representative samples of both different load and defect widths.

Table 3.1 – The data characteristics of each scenario

Scenarios	Signal Length	Defect Width ($in.$)		Motor Load (hp)	
	(No. of Segments)	$0.007in.$	$0.014in.$	$0hp$	$1hp$
SCN 1	102400 (100)	✓	-	✓	-
SCN 2	102400 (100)	✓	-	-	✓
SCN 3	102400 (100)	-	✓	✓	-
SCN 4	102400 (100)	-	✓	-	✓

3.2 A Brief Study on Semi-Supervised Dimensionality Reduction

The proposed scheme for the fault diagnosis system is illustrated in Fig. 3.1. This module initially segments the vibration signal into various non-overlapping partitions and, then, passes them through the WPT. It decomposes each segment into 16 different equal size packs with default frequency sub-bands as results of four levels

of decomposition. Finally, seven different statistical features including Root mean square, Skewness, Kurtosis, Maximum, Minimum, Peak to peak and Variance of each pack are extracted (i.e., $16 \times 7 = 112$ features).

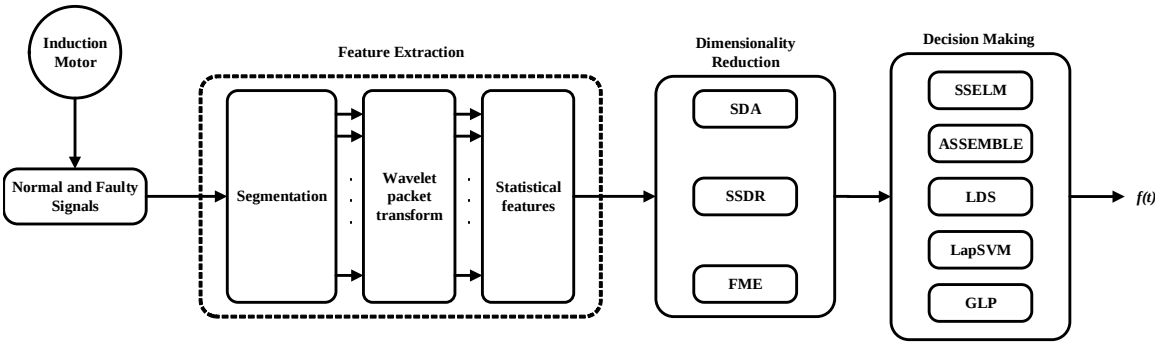

Figure 3.1 – The general scheme of the diagnostic system including feature extraction, dimensionality reduction and decision making module. [1]

Then, the 112 processed features are fed to the next module, dimensionality reduction, which provides the proper sets of small size features for the decision making module.

3.2.1 Results

After the dimensionality reduction, generated features by each technique are fed to five semi-supervised algorithms: SSELM, ASSEMBLE, LDS, LapSVM, and GLP (see Fig. 3.2). The aim is to determine the efficiency of the selected algorithms for bearing fault diagnosis in IMs.

The pre-processed data obtained by SDA and FME are embedded into four-dimensional feature space. Different embeddings for SSDR are examined and the one with the best accuracy, which is transformation into three dimensional feature space $(n_\Omega - 1)$, is considered in this paper.

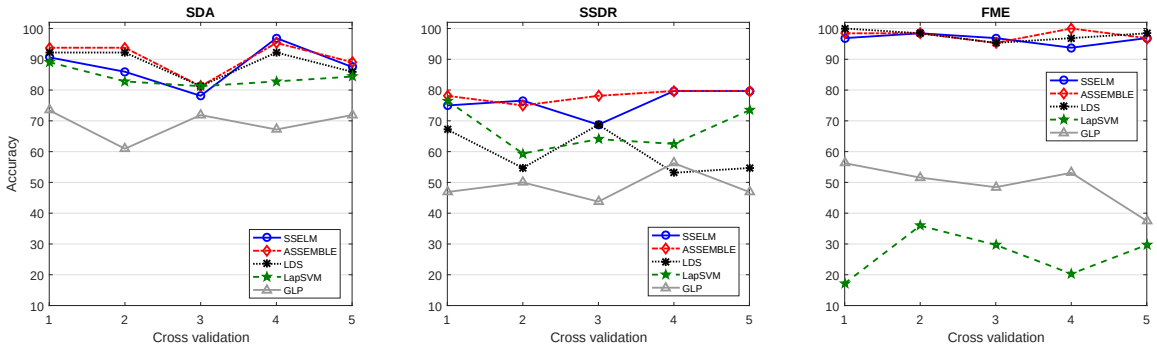

Figure 3.2 – The accuracy of each classification algorithm over cross validation iterations, using different dimensionality reduction techniques.

Accuracy

In Fig. 3.3, it is observable that FME outperformed the others while it is used along with SSELM, ASSEMBLE, and LDS. This is while LapSVM and GLP are not accurate enough on the FME output. The main reason behind the attained accuracy by GLP and LapSVM is that they are both binary classifiers, and used in a one-versus-all approach, which has degraded their performance, specially on FME output. Conversely, SDA is providing suitable features for such classifiers utilizing the one-versus-all approach. On the other hand, SSDR is providing classifiers with features that result in a moderate accuracy.

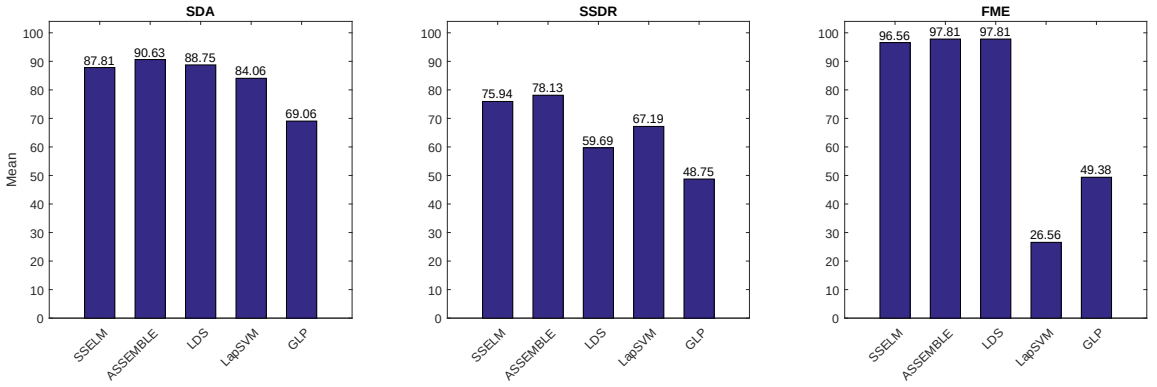

Figure 3.3 – The mean performance attained by each classifier over the cross validation on different dimensionality reduction techniques.

Aside from the influence of the dimensionality reduction module, we study the performance of the classification algorithms for diagnosing bearing defects in IMs. For instance, GLP is extremely dependent on the initialization phase, where it constructs a graph on the data. Providing its intrinsic harmonic function with a fully connected yet accurate graph on the data is a delicate step since, even using optimization, sometimes the constructed graph is not satisfying. Another example would be LapSVM, which assumes that observations of different classes are distant from each other. Regardless of its kernel type, violation of this assumption will effect the results as the estimated manifolds through the manifold learning phase may not be accurate. Notice that the aforementioned issues could be less critical in a binary problem. This is while the other three fault classifiers, specially ASSEMBLE, seem to be more flexible. ASSEMBLE is highly adaptable to more complicated decision boundaries due to the nature of the boosting algorithm.

It is worthwhile to mention that the use of the one-vs-one approach results in a higher accuracy than the one-vs-all approach, since the created ambiguous area would be much smaller. Nevertheless, usually using one-vs-one is not preferable, since it is computationally very expensive. Furthermore, one-vs-all method is more commonly used for SVM family of classifiers, including LapSVM [53].

Stability

Considering Fig. 3.4, SSELM, ASSEMBLE, and LDS highest rate of stability is resulted by FME. However, LapSVM is very stable with the SDA features. GLP, on the other hand, has almost similar range of stability in all experiments, which reaches its best rate on SSDR features. Thus, it can be concluded that FME is providing the best set of features for the semi-supervised multiclass fault classifiers in this study, and SDA is the best choice for the semi-supervised binary fault classifiers.

According to Table 3.2, the choice of the most stable classifier, should be made among SSELM, ASSEMBLE, and LapSVM since they best match with FME, SSDR,

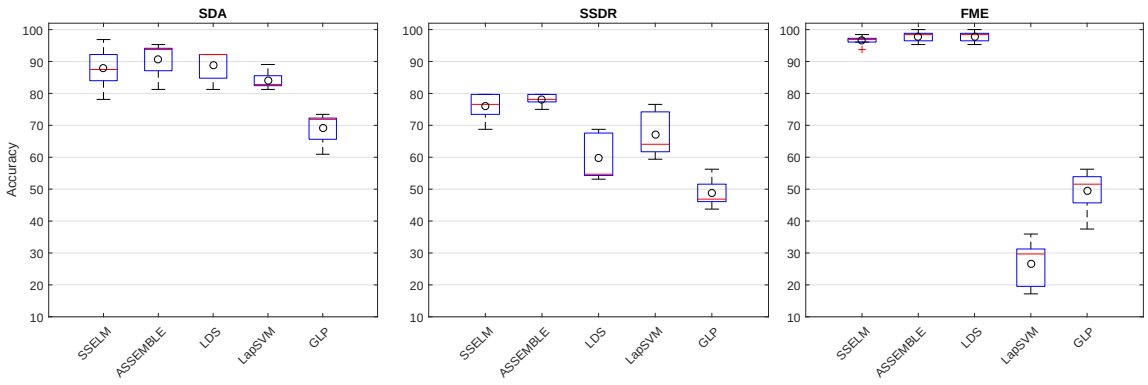

Figure 3.4 – The distribution of classification accuracy achieved by each classifier through each dimensionality reduction technique.

Table 3.2 – The standard deviations attained by fault classifiers through the cross-validation. The winners are specified by bold font.

	SSELM	ASSEMBLE	LDS	LapSVM	GLP
SDA	6.58	5.74	4.99	**3.01**	5.11
SSDR	4.50	**1.91**	7.61	7.41	4.74
FME	**1.71**	1.78	1.78	7.65	7.21

and SDA, receptively. However, considering their results through the entire experiments, ASSEMBLE can be considered as the most stable algorithm in this study. It has the highest stability using SSDR, and almost as stable as SSELM using FME, with an ignorable difference. Although ASSEMBLE ranked third in SDA series of experiments, its stability is still acceptable and comparable with the others.

Discussion

Fig. 3.5 shows the the result of each DR technique. It can be seen that SDA and FME resulted in the best feature spaces. Considering the attained performance, the combination of FME and ASSEMBLE is followed by high stability and accuracy for this experiment. Although FME is the winner in this experiment, it may not be possible to conclude that it is the best DR technique. FME seems to be very compatible with some algorithms and highly incompatible with some other algorithms. SDA, on

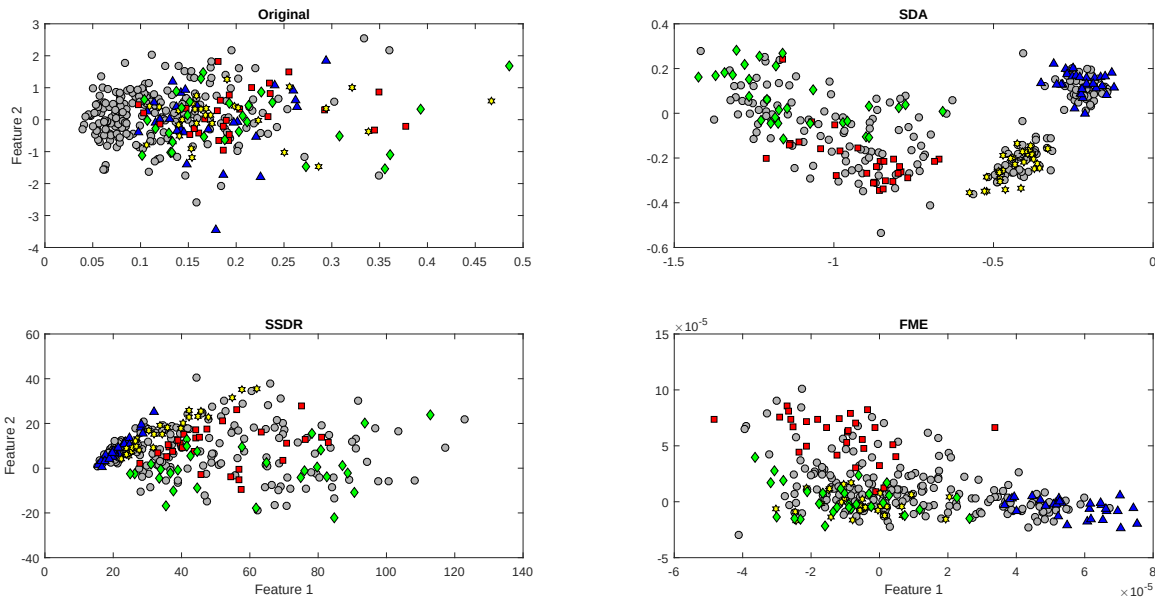

Figure 3.5 – The distribution of the data observations before the dimensionality reduction (Original) and after dimensionality reduction using SDA, SSDR, and FME.

the other hand, is more reliable in this sense due to the fact it provides acceptable performance for most of classifiers the classifiers. Thus, SDA may be a robuster DR technique for a diagnostic system.

Note that in this work, FME projects the given features to a n_Ω (number of classes) dimensional feature space. Therefore, as the number of fault classes increases, the dimensionality of the DR module increases as well. In case of employing different fault classifiers other than those suggested in this paper, where the aforementioned condition applies, one should make sure that the chosen fault classifier is able to work properly on high dimensional data as well.

The choice of the base learner for ASSEMBLE in this work is k-nearest neighbor with $k = 3$ as it resulted in the best performance for the target application. However, other base learners such as decision tree are used as well and resulted in almost the same performance with a slight difference.

3.3 Semi-Supervised Smooth Alpha Layering

The main contribution of this chapter is to propose a novel semi-supervised algorithm, called Semi-supervised Smooth Alpha Layering (S3AL), which aims to perform multi-class classification by resorting to an inductive learning procedure. In this algorithm, the structure of data is captured using α-Shape [13] instead of graph estimation in the manifold learning. By doing so, the structural shape of the data is estimated without the need for multiple parameters tuning. α-Shape is a strong tool, which is mostly utilized for the surface estimation. However, to our knowledge, it is rarely exploited for SSL and designing a FDS.

3.3.1 Overview of α-Shape

α-Shape is indeed a generalization of the convex hull. Supposing $X = U \cup L = \{x_1, x_2, ..., x_n\}$ is a set of all data samples, where x_i is defined by a vector $\{x_{i1}, x_{i2}, ..., x_{id}\}$ in $\mathbb{R}^d$, the convex hull of X, Conv(X), is defined as the smallest convex set of X. To construct the α-Shape of X, $\mathcal{A}(X)$, first Conv(X) should be estimated, so that the attained structure could be shrunk further to reach the best shape. In other words, the convex hull of the samples can be considered as a shattered crystal that its pieces are held together. This convex hull embraces all samples. To obtain the most representative shape, some of those shattered pieces have to be removed from the crystal in a way that the remained pieces resemble the shape of data samples, which is in fact $\mathcal{A}(X)$. In mathematics, those shattered pieces are known as simplexes. A simplex is a geometrical shape resulted from the Conv(Δ), where $\Delta = \{x_i\}_{i=1}^{d+1}$. The α-Shape construction begins with estimating Conv(X), which is basically obtained by the Delauney triangulation of $\{X \in \mathbb{R}^d \mid d \leq 3\}$. In general, the Delauney tessellation, which is an extention of the Delauny triangulation, can be used for the higher-dimensional data $d > 3$. The aim of the Delaunay tessellation is to produce a set of simplexes $\Lambda = \{\Delta_1, \Delta_2, ..., \Delta_m\}$ on the data in a way that none of the samples in X lies in the

circumsphere of any $\Delta_\gamma \subset \Lambda$. Once the tessellation is performed, the total convex hull can be obtained by: $\text{Conv}(X) = \{\bigcup_{\gamma=1}^{m} \text{Conv}(\Delta_\gamma) : \Delta_\gamma \subset \Lambda\}$.

The desirable tessellation can be obtained using different techniques such as Quickhull [54], which is used in this study.

Given that $\mathcal{A}(X) \subset \text{Conv}(X)$, it shrinks the achieved convex hull to reach the desired shape. To do so, any simplex that does not fit in a sphere (assuming $d = 3$) with the radius of α will be considered for removing its unshared faces. Therefore, the level of details of an α-Shape has an inverse relationship with the parameter α. In other words, the procedure is similar to carving the convex hull without removing any $x \in X$, by means of a d-dimensional rounded geometrical shape (e.g., a sphere for $d = 3$), with α radius. Then, the remained simplexes resemble the target α-Shape.

Although capturing the data structure using α-Shape helps to improve the classification accuracy, it has been rarely used for SSL. This might be due to the following challenges: (a) overlapping classes can further complicate the process of α-Shape construction, since it requires completely separable classes; (b) a particular value of α is not sufficient to construct discriminant shapes for the classes with different densities. To address these issues, this chapter proposes a novel SSL strategy.

3.3.2 Algorithm Description

S3AL consists of two major phases as detailed in Algorithm 1 and 2. The first phase easily classifies datasets compatible with any SSA. It only classifies those samples located in the safe zones (i.e., regions in which the data can be classified with a high confidence) and send overlapping samples and outliers into the second phase. Each phase is formally explained in the following:

Phase 1

To capture the shape of classes with different densities, we firstly extract the safe zones of each class in an iterative process. In each iteration, the α radius changes on a spectrum Ξ, which contains all important values of α (i.e., values that change the α-Shape). To begin the procedure, for each simplex Δ_γ in the resulted Delauney tessellation $\Delta_\gamma \subset \Lambda$, a different value of α is generated to form Ξ as follows:

$$\forall \Delta_\gamma \subset \Lambda : \begin{vmatrix} x_{01}+x_{02}+\cdots+x_{0d} & x_{01} & x_{02} & \cdots & x_{0d} & 1 \\ x_{11}+x_{12}+\cdots+x_{1d} & x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{d'1}+x_{d'2}+\cdots+x_{d'd} & x_{d'1} & x_{d'2} & \cdots & x_{d'd} & 1 \end{vmatrix} = 0 \tag{3.1}$$

where $d' = d+1$; $\{x_{0j}\}_{j=1}^{d}$ are unknown; and $\{x_{ij}\}_{i=1}^{d'}$ are the vertices of Δ_γ. For the ease of explanation, assume that $d = 3$. Then, by expanding the determinant:

$$\varpi(x_{01}^2 + x_{02}^2 + x_{03}^2) - (D_1 x_{01} + D_2 x_{02} + D_3 x_{03}) + \beta = 0 \tag{3.2}$$

where

$$\varpi = \begin{vmatrix} x_{11} & x_{12} & x_{13} & 1 \\ x_{21} & x_{22} & x_{23} & 1 \\ x_{31} & x_{32} & x_{33} & 1 \\ x_{41} & x_{42} & x_{43} & 1 \end{vmatrix}, \tag{3.3}$$

$$\beta = \begin{vmatrix} x_{11}+x_{12}+x_{13} & x_{11} & x_{12} & x_{13} \\ x_{21}+x_{22}+x_{23} & x_{21} & x_{22} & x_{23} \\ x_{31}+x_{32}+x_{33} & x_{31} & x_{32} & x_{33} \\ x_{41}+x_{42}+x_{43} & x_{41} & x_{42} & x_{43} \end{vmatrix}, \tag{3.4}$$

and $\{D_j\}_{j=1}^{3}$ is the determinant (negative for even j values and positive for odd ones) attained from D, in which its $1+j$ column is discarded:

$$D = \begin{bmatrix} x_{11}+x_{12}+x_{13} & x_{11} & x_{12} & x_{13} & 1 \\ x_{21}+x_{22}+x_{23} & x_{21} & x_{22} & x_{23} & 1 \\ x_{31}+x_{32}+x_{33} & x_{31} & x_{32} & x_{33} & 1 \\ x_{41}+x_{42}+x_{43} & x_{41} & x_{42} & x_{43} & 1 \end{bmatrix} \tag{3.5}$$

The circumsphere of Δ_γ is then achieved by completing the square:

$$\varpi\left(x_{01}-\frac{D_1}{2\varpi}\right)+\varpi\left(x_{02}-\frac{D_2}{2\varpi}\right)+\varpi\left(x_{03}-\frac{D_3}{2\varpi}\right)=\frac{D_1^2+D_2^2+D_3^2}{4\varpi}-\beta=\alpha^2 \tag{3.6}$$

The radius is then computed and added to the spectrum:

$$\Xi=\bigcup_{i=1}^{m}\alpha_i \ , \alpha=\frac{\sqrt{D_1^2+D_2^2+D_3^2-4\varpi\beta}}{2|\beta|} \tag{3.7}$$

α_i values are then sorted in a descending order. To capture the densest class, an α-Shape is constructed by means of the largest α value in Ξ, and, then, the level of detail is iteratively increased by reconstructing through the next α value in Ξ until the α-Shape becomes pure, i.e., all the embraced labeled samples belong to the same class (lines 4-15 in Algorithm 1). Then, all samples in the attained α-Shape S are stored as a separate shape in Ψ (lines 9 and 10 in Algorithm 1), which is initially set to $\Psi=\varnothing$, and discarded from the pool Φ (line 16 in Algorithm 1). The pool is initially set to $\Phi=X$. This procedure is repeated until a stopping criterion has been met (lines 1-17 in Algorithm 1). This algorithm stops whenever $\text{Card}(\Phi)<d+1$, where Card(.) stands for the cardinality.

In this procedure, α-shapes with more than one region $S=\{r_i\}_{i=1}^{n_r}$, in which n_r stands for the number of regions in S, might be created as the value of α decreases. Each region r_i is then stored as a separate α-Shape $\mathcal{A}(r_i)$ (lines 9 and 10 in Algorithm

1). Besides, some of these newly created shapes may contain only unlabeled samples, and, thus, they left unlabeled and stored in Ψ_u that is initially set to $\Psi_u = \varnothing$.

The data is now divided into multiple α-Shapes. Samples in each of these stored α-shapes are being labeled, called pseudolabels, by means of the labeled samples in these shapes. Then, the high confidence samples whose pseudolabels are equal to their affinity measure are extracted in order to form a safe zone (lines 19-23 in Algorithm 1). To compute the affinity measure q_i, first the affinity matrix is computed as follows:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right), \tag{3.8}$$

where σ is set to the standard deviation of $\bigcup_{i,j=1}^{n} \|x_i - x_j\|$. Then, assuming that $L = \{l_1, l_2, ..., l_{n_\Omega}\}$ and l is the set of all labeled samples with an identical y, q_i is calculated as:

$$\forall x_i \in U \wedge x_j \in l_\kappa : q_i = \underset{\vartheta_\kappa, 1 \leq \kappa \leq n_\Omega}{\arg\max} \sum_{j=1}^{n_\kappa} W_{ij}, \tag{3.9}$$

where n_κ is the number of elements in l_κ. The rest of the data samples are kept aside in a residual set R for further processing in the next phase (line 21 in Algorithm 1).

Phase 2

Assuming that the final α-Shape contains different regions, where each region is indeed the α-Shape of an individual class, the predictions should be smooth over each region. The predictions from the safe zones are most likely to be smooth over their shapes. Here, smoothness assumption is locally applied to the residual set $x_i \in R$, in an iterative manner, that aims to generate pseudolabels layer by layer (lines 29-33 in Algorithm 2). To do so, at each iteration, smoothness is applied on layers of samples near the labeled safe zones, that are extracted in the first phase. Then, the smoothed layers is added to the related safe zone. The procedure continues until the algorithm assigns a pseudolabel $\hat{y}_i$ to every $x_i \in R$. In fact, the final α-Shapes are attained by

Algorithm 1: S3AL (phase 1)

Input: X, base learner f
Output: Classification model h, Y
Definition:
$\lceil . \rceil$ returns unique values
n_c is the maximum number of classes in the α-Shape.
Initialization:
Compute affinity matrix W using (3.8)
Create the pool $\Phi = X$
Create the shape repositories $\Psi = \Psi_u = \varnothing$
Create the residual set $R = \varnothing$

```
while Card(Φ) ≥ d + 1 do
    Form the spectrum Ξ using (3.7)
    Set α = max(Ξ)
    while true do
        Estimate the α-Shape: S = A(Φ)
        Given that Y_i = {y_j ∈ r_i}_{j=1}^{n_y(i)}, where n_y(i) is the number of labels in
          r_i, return n_c for S: ∀r_i ⊂ S: n_c = max Card(⌈{Y_i}_{i=1}^{n_y(i)}⌉)
        if n_c ≤ 1 then
            for ∀r_i ⊂ S do
                Assign ⌈{Y_i}_{i=1}^{n_y(i)}⌉ to r_i
                Store r_i: Ψ = Ψ ∪ A(r_i)
            end for
            break
        end if
        α ← next α in Ξ
    end while
    Remove the stored samples: Φ = Φ − S
end while
∀x_i ∈ U: compute the affinity measure q_i by (3.9)
for ∀S_j ⊂ Ψ, j = 1, ..., Card(Ψ) do
    if ∃ y_i ∈ S_j then
        ∀y_i ≠ q_i: remove the assigned y_i then R = R ∪ x_i
        Reshape the α-shape with the confident samples:
          S_j ← S_j = A({∀x_i | y_i = q_i})
    else
        Ψ_u = Ψ_u ∪ S_j
        Ψ = Ψ − S_j
    end if
end for
Add the unlabeled shapes to the residual: R = R ∪ Ψ_u
```

Algorithm 2: S3AL (phase 2)

29 **while** $R \neq \varnothing$ **do**
30 | $\forall x_i \in R$ generate a pseudolabel $\hat{y}_i$ using (3.10)
31 | Add $(x_i, \hat{y}_i)$ to the shape with the class $\hat{y}_i = \vartheta_\kappa$
32 | Remove x_i from residual: $R = R - x_i$
33 **end while**
34 $\forall S_j \subset \Psi_u,\ j = 1, ..., \mathrm{Card}(\Psi_u)$: reassign a unique pseudolabel $\hat{y}_j$ to all samples of S_j by voting:

$$\hat{y}_j = \underset{\vartheta_\kappa, 1 \leq \kappa \leq n_\Omega}{\arg\max}\ \mathrm{Card}(\{\forall y_i \in S_j \mid y_i = \vartheta_\kappa\})$$

35 Find x_i in the overlapping regions using (3.11) and remove their pseudolabels and the add them to $R = R \cup x_i$
36 $\forall x_i \in R$: x_i receives $\hat{y}_j$ of its nearest α-Shape
37 Store the fixed predictions in $\hat{Y}$
38 Form a model $h = f(\hat{Y})$

expanding the safe zones, layer by layer. $\hat{y}_i$ for each sample is calculated as:

$$\forall x_i \in R \wedge x_j \in N(x_i) : \hat{y}_i = \underset{\vartheta_\kappa, 1 \leq \kappa \leq n_\Omega}{\arg\max} \sum_{j=1}^{n_\kappa^*} W_{ij}, \tag{3.10}$$

where $N(x_i)$ is the neighbourhood of x_i in which the number of labeled samples representative of class κ is denoted by n_κ^*. Thus far, a pseudolabel $\hat{y}_i$ is assigned to each sample by choosing the most similar class in the neighbourhood $N(x_i)$. Then, the pseudolabels in each shape $S_j \subset \Psi_u$ are smoothed by assigning the label of the majority class in S_j to its samples (line 34 in Algorithm 2).

Here, the neighbourhood N is estimated by k nearest neighbours, where k is set to 15. Although the value of k can be specified by user as a free parameter, an odd value in the range of $\{k = 2\varepsilon + 1 : 5 \leq \varepsilon \leq 9\}$ is preferred for this case study. To cancel the impact of selecting an improper k, the algorithm looks for any overlap between α-Shapes of individual classes. An overlap implies that the smoothness does not hold around the decision boundary. An overlapping region can be approximated, then, by

finding the samples located in the circumsphere of a simplex. This can happen when α-shapes are too close or have intersects. Assuming $d = 3$ and $\forall x_i = \{x_{i1}, x_{i2}, x_{i3}\}$, a sample x_i is in the circumsphere of a simplex if:

$$\varpi \left(x_{i1} - \frac{D_1}{2\varpi} \right) + \varpi \left(x_{i2} - \frac{D_2}{2\varpi} \right) + \varpi \left(x_{i3} - \frac{D_3}{2\varpi} \right) < \alpha^2 \tag{3.11}$$

If an overlap is detected, all the samples inside that overlap are re-labeled based on the nearest α-Shape to relax the decision boundary (lines 36 and 37 in Algorithm 2).

3.4 Design of the Diagnostic System

The proposed hybrid diagnostic framework is illustrated in Fig. 3.6. This hybrid framework contains five steps as follows:

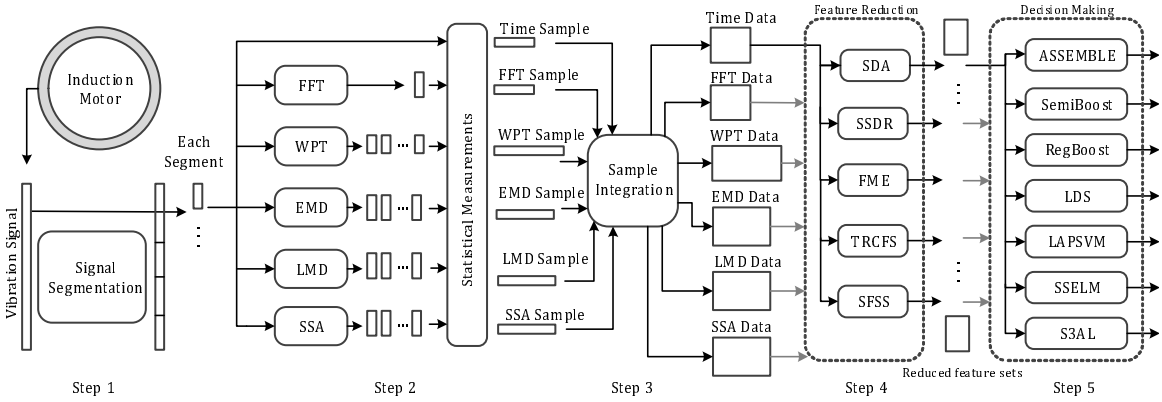

Figure 3.6 – The designed hybrid framework, where the FR and decision making steps are specified by the dashed boxes.

3.4.1 Signal Segmentation

The diagnostic procedure begins with segmentation. In Fig. 3.6 the collected vibration signals are firstly passed through the signal segmentation step, in which the input

signals are divided into a number of non-overlapping parts with an equal length. This helps in transition of a signal into several segments that are fed to the next step for feature extraction. Each segment is sent to each of the five FE approaches.

3.4.2 Feature Extraction

Time-Domain analysis reveals the statistical characteristics of the segments in the time-domain. It extracts and returns a feature vector V including 14 statistical measures [8], i.e., V=[Maximum, Minimum , Peak to peak, Mean value, Variance, Root mean square, Skewness, kurtosis, Crest factor, Impulse factor, Shape factor, Margin factor, Entropy, Energy] from each segment, to construct a time-domain feature set.

Fast Fourier Transform (FFT) has been extensively used in fault diagnosis to obtain the frequency characteristics of the signals. In this paper, the frequency spectrum of each signal is achieved by applying FFT, and, then, feature vector V is calculated over the obtained frequency-domain spectrum.

Regarding to the time-frequency analysis, three state-of-the-art approaches are used to decompose non-stationary signals, including the Wavelet Packet Transform (WPT) [55], the Empirical Mode Decomposition (EMD) [56] and the Local Mean Decomposition (LMD) [57].

Moreover, a singular spectrum analysis, which makes use of singular value decomposition (SVD) for time-series analysis is used to form a time-domain feature set[57]. Fig. 3.6 shows that as the raw vibration signal traversed from the left to the right steps, more informative feature sets are being collected. Hence, the best way for diagnosing bearing defects belongs to the best possible combination of the approaches from each step that can provide the most useful information to construct the SSL learners.

3.4.3 Sample Integration

Each FE approach receives signal segments, and, then, it generates a set of features for each segment, and, consequently, calculates statistical measures for each individual feature of the set. Given a signal segment, this results in producing only one sample with a larger number of features. The corresponding high-dimensional samples are being collected from all segments, and, then, integrated together to form a high-dimensional feature set.

3.4.4 Feature Reduction

The optimal size of the resulted feature sets from the previous step, is then individually estimated using each of the semi-supervised FS and DR approaches namely Semi-supervised Discriminant Analysis (SDA) [34], Semi-Supervised Dimensionality Reduction (SSDR) [35], Flexible Manifold Embedding (FME) [36], Trace Ratio Criterion Feature Selection (TRCFS)[37] and Structural Feature Selection with Sparsity (SFSS) [38]. These approaches are among the state-of-the-art semi-supervised feature reduction algorithms and used in this step for the sake of a comprehensive comparison.

Among the selected DR approaches, SDA captures the geometric structure of the data using U, while the between-class separability is maximized in the new feature space using L. SSDR, on the other hand, calculates the projection matrix based on two constraints indicating whether a pair of samples in the original input space belong to the same class or not. FME integrates the smoothness on the data manifold and the label fitness, in order to achieve an optimal projection matrix.

Moreover, this step makes use of two state-of-the-art semi-supervised algorithms for FS. TRCFS finds the best features through a filter-based approach. Using this approach, the within-class and the between-class scatter matrices are formed on the soft labels estimated by the label propagation, and the best features are then selected w.r.t. a noise insensitive trace ratio criterion. On the other hand, SFSS aims to select

the features jointly, while the manifold learning is utilized to perform SSL on U.

3.4.5 Decision Making

The last step of the proposed diagnostic framework is the decison making, which receives multiple inputs including all reduced feature sets that are obtained through each possible combination of FE and FR approaches. This step includes multiple state-of-the-art approaches for SSL including Adaptive Semi-Supervised Ensemble (ASSEMBLE) [44], Semi-supervised Boosting (SemiBoost) [45], Regularized Boosting (RegBoost) [33], LDS [41], LapSVM [31], SSELM [42], and the proposed semi-supervised learner S3AL. ASSEMBLE iteratively maximizes the pseudo-margin. In SemiBoost, the pseudomargin is maximized w.r.t. both cluster and manifold assumptions, where a graph-based similarity matrix is leveraged to iteratively update the hypothesis space based on the most confident samples. RegBoost clusters the data in order to generate initial pseudolabels. It then maximizes the margin, while the decision boundary is regularized using SSA. LDS maximizes the margin on a low-dimensional data attained by a graph-based DR procedure. LapSVM maximizes the margin, while minimizing the inconsistency on the data manifold structure. SSELM exploits manifold regularization to adapt Extreme Learning Machines for SSL.

3.5 Experimental Results

Experimental setting is firstly explained in this section. The attained results are then analyzed and compared.

3.5.1 Experimental Setting

Initially the collected signal for each scenario is segmented and fed to the FE step. After the sample integration step, the extracted features are then fed to the FR

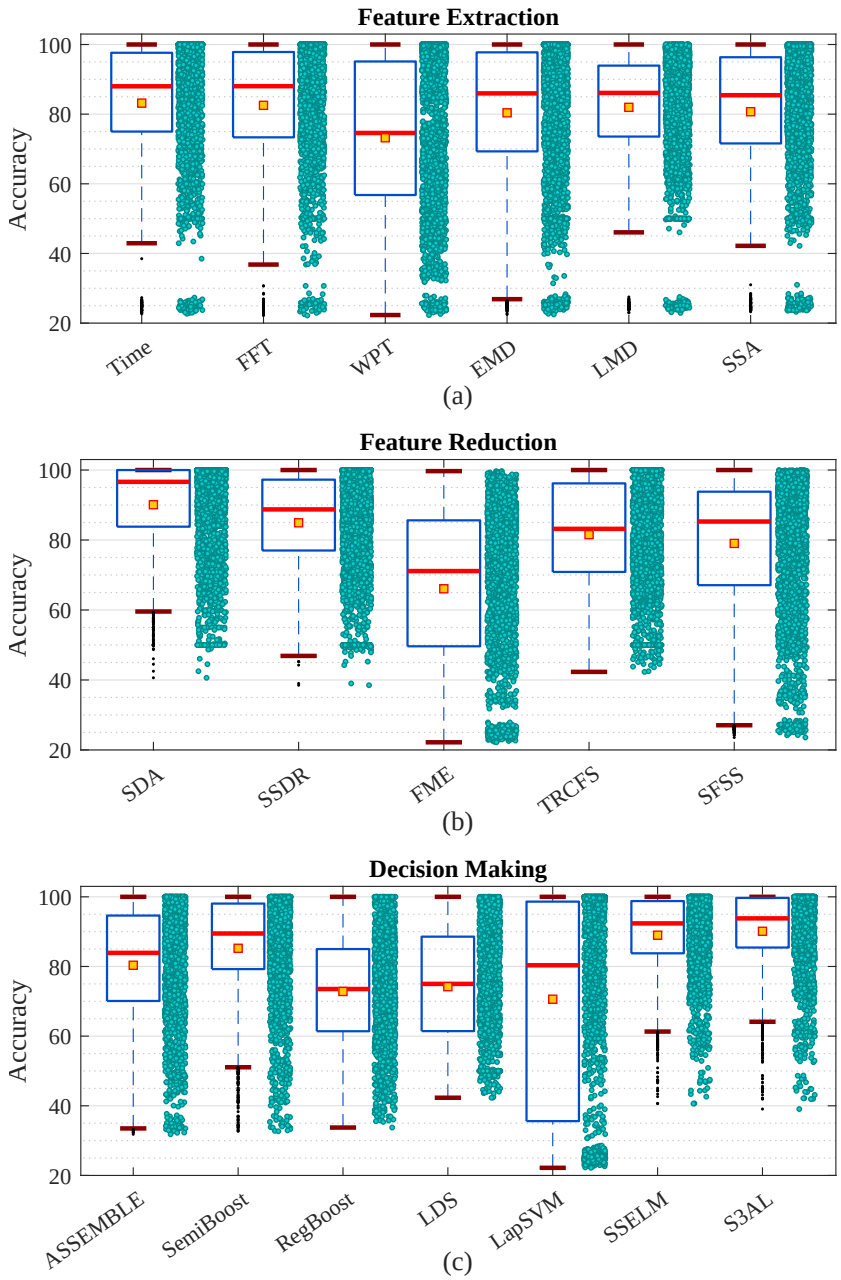

Figure 3.7 – Distribution of the achieved accuracies by each approach on all experiments with different rates, where each panel (a), (b) and (c) focuses on presenting the accuracies achieved through different approaches of FE, FR and decision making steps, respectively.

and the decision making steps. To evaluate our experiments, a 5-fold nested cross-

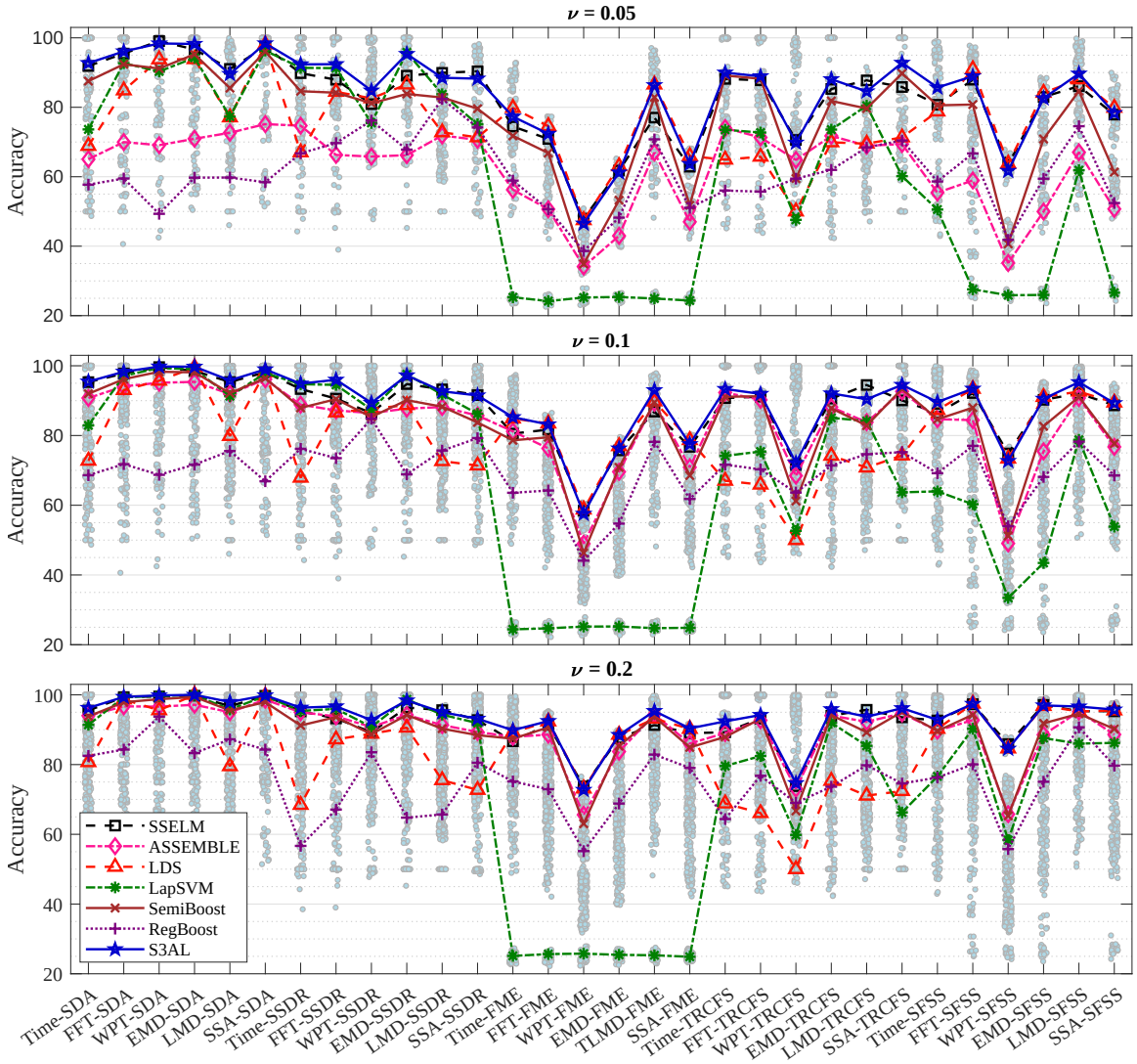

Figure 3.8 – The achieved accuracy by the combination of each SSL approach and each pair of FE and FR approaches. The achieved accuracies by each SSL approach are averaged over all possible experiments at each rate v; 5%, 10% and 20%.

validation scheme has been used. Using this technique, cross-validation is performed by means of an inner and an outer loop. Firstly, the acquired data for each experiment is equally divided into five folds. Then, the outer loop is repeated five times, where each time one fold is kept for testing and the rest forms the training set. The training set is split into L and U by using a random stratified selection. Then, the training set will go through the inner loop for the parameter estimation. To do so, the training

set goes through another cross-validation procedure, in which one fold is kept for validation and the other three form the inner training set. In this procedure, each iteration of the outer loop is further repeated five times, where each time different L and U are randomly selected for a training set w.r.t. a rate $\nu = \frac{|L|}{|U+L|}$. Here, three values of 5%, 10% and 20% are considered for ν.

The proposed algorithm, S3AL, and boosting algorithms employed in this chapter such as RegBoost, SemiBoost, and ASSEMBLE benefit from a base classifier. In all experiments, the decision tree is employed as a base classifier. Besides, binary algorithms, in this work, are converted to multiclass classifiers using the one-versus-all strategy.

3.5.2 Results

Fig. 3.7 illustrates a statistical comparison of the state-of-the-art approaches at each step in classifying bearing defects in the induction motor over all possible experiments. Each panel of this figure represents the distribution of all attained accuracies (solid circles) attained by each combination (FE approaches $\times$ FR approaches $\times$ SSL approaches $\times$ experiments). Fig. 3.7(a) illustrates the distribution of all accuracies achieved by each FE approach. The boxes show the distribution range of these accuracies among 1st and 3rd quartiles, solid squares show the average of all attained accuracies by each FE approach, solid lines in the boxes show the median of the achieved accuracies by each FE approach, dash lines and small dots illustrate the outlier range and the outliers, respectively.

It can be seen from the Fig. 3.7(a) that different choices of FE approach does not significantly change the attained accuracies. However, extracted features by means of Time analysis can result in the highest accuracies. Also, LMD has the lowest standard deviation. This is while, WPT results in a wider range of accuracies, which indicates that not every combination of FR and SSL approaches with WPT can lead to a high accuracy.

Fig. 3.7(b) illustrates the distribution of the archived accuracies by each semi-supervised FR approach. The choice of the FR approach is very important as the accuracy is highly dependant on the generated features by each FR approach. In fact, the reason for lower sensitivity to the FE approaches is that the FR approaches drastically transform the samples, where the effect of the original features can be neglected. Based on Fig. 3.7(b), SDA results in the best accuracies and the lowest standard deviation. This is while FME seems to be very sensitive to its combination with other approaches, and yields unsatisfactory accuracies in some cases.

Fig. 3.7(c) shows the distribution of the achieved accuracies through each SSL approach. These contain all the achieved accuracies by all FE and FR approaches. The SSL approaches are ranked based on the averaged accuracies over all experiments as S3AL, SSELM, SemiBoost, ASSEMBLE, LDS, RegBoost and LapSVM. S3AL shows the minimum variation and outperforms other SSL approaches. LapSVM is the least stable approach, which yields the largest number of outliers.

To study the accuracy of each combination, Fig. 3.8 shows the overall accuracies of the SSL approaches along with FE and FR approaches in different rates. Different panels of the Fig. 3.8 shows that S3AL attains the best accuracies in most of the combinations for all rates, while SSELM stands for the second rank. From the stability point of view, S3AL, SSELM, and SemiBoost seem to be less sensitive to the combination with FE and FR approaches, while LapSVM, LDS and RegBoost show a higher sensitivity to the choice of FE and FR approaches for combinations.

By analyzing the behaviour of the SSL approaches, it can be seen that accuracy improves as the value of ν increases. Combinations of LapSVM and FME are very incompatible, since the attained accuracies through their combination are dropped and remained unchanged over different rates.

Considering all possible rates and combinations, the maximum accuracies are mostly achieved using SDA. However, combinations of LDS and RegBoost with SDA do not result in best accuracies. FME and SFSS show high dependency to the ratio,

and they are not the best choice for the lowest ratio.

There exist a slight difference among possible combinations of SDA and FE approaches in terms of accuracy. Nevertheless, EMD and WPT result in the maximum accuracy. Top rank accuracies are mostly obtained by S3AL. Although the difference is insignificant, WPT is ranked first among the FE approaches, since its combinations are resulted among the top rank accuracies most often.

The most accurate combination is achieved through linking WPT, SDA and S3AL. Although the averaged accuracies of WPT over all combinations is not among the top FE approaches as shown in Fig. 3.7, the maximum accuracy is achieved by the combination WPT+SDA+S3AL.

3.6 Summery

In this chapter, firstly a comparative study is performed on semi-supervised DR techniques through designing a diagnostic framework. Then, a hybrid diagnostic framework is introduced, which benefits from the semi-supervised learners. They enable the hybrid framework to generate a diagnostic model, where the initial data are not fully labeled and a large number of samples are collected with missing labels. This framework is applied for classifying bearing defects in IMs. It has three important steps for feature extraction, feature reduction and decision making. To improve the diagnostic accuracy of the proposed framework, a novel SSL approach, called S3AL, is proposed that uses the concept of α-Shape. The obtained accuracies indicate that S3AL outperforms other state-of-the-art SSL rivals in all experiments with different rates. In addition, the proposed framework facilitates an extensive comparison for each step through four distinct scenarios with different settings and rates. This comparative study also reveals the effects of each state-of-the-art FE and FR approaches in terms of diagnostic accuracy, and determines the best possible combination of the approaches in the hybrid framework.

Chapter 4

Fault Diagnosis in High-dimensional Feature Space with Limited Supervision

The good performance of a data-driven fault classifier can be guaranteed by resorting to a proper set of input features. Indeed, the input features may contain redundant and non-informative features. This results in devising feature selection or reduction components in the design of the diagnostic systems [15]. On the other hand, there has been an increasing interest on the use of deep neural networks (DNN) in recent years, as a result of the increased dimensionality of the data and the growing demand for higher classification performance. Multiple hidden layers of a DNN enable feature extraction on the data, where at each layer, more abstract features are extracted by means of the outputs of the preceding layer. This multi-level feature extraction procedure substitutes the feature reduction procedure, where fewer but more representative features are created from the pool of features. Here, a semi-supervised DNN has been adopted for fault diagnosis, which eliminates the need for devising an extra phase of feature reduction in the FDS framework and further improves the diagnostic efficiency.

The proposed diagnostic system in this chapter is followed by various contributions. The main contribution of this work is to design a hybrid framework for diagnosing gear faults in a very high dimensional feature space, in which only a few number of labeled samples are collected along with a large number of unlabeled samples. The proposed FDS system has two major components. The first one contains several advanced feature extraction methods and the other contains an advanced semi-supervised deep learning method, so-called Semi-Supervised Deep Ladder Net-

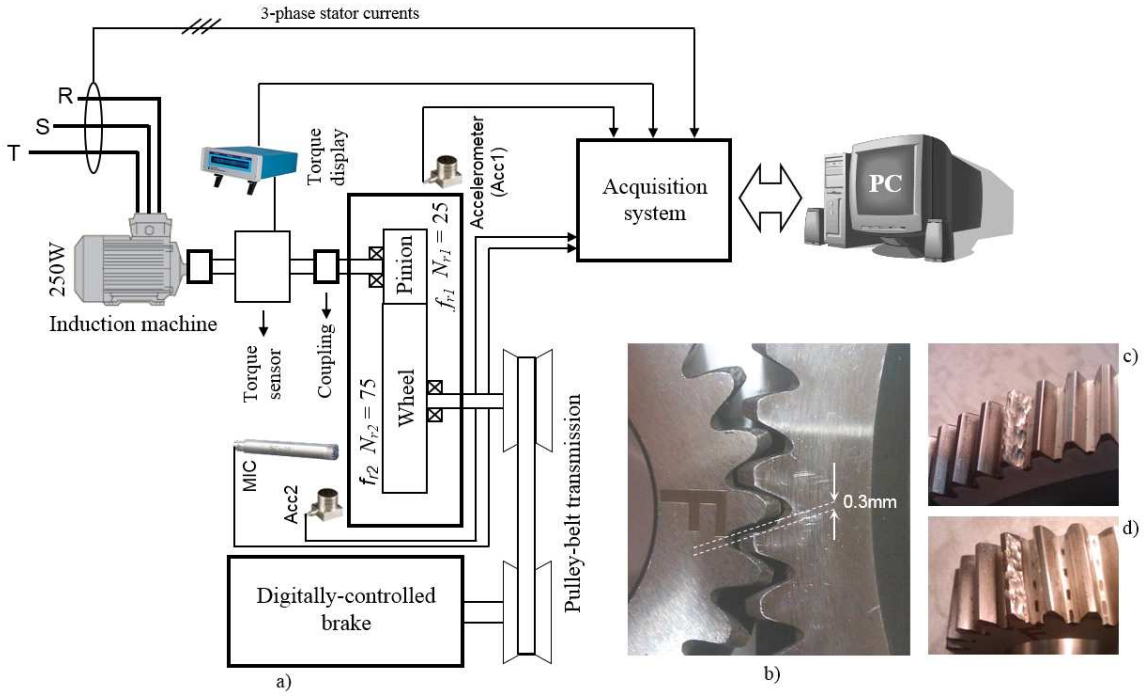

Figure 4.1 – a) Electromechanical system under study, b) Pinion-wheel contact at the damage point, c) Wheel tooth damage and d) Pinion tooth damage.

work (SSDLN) [58], for decision-making. Besides, various advanced semi-supervised and supervised networks have been devised in this hybrid framework. This facilitates a comparative study between the advanced SSL methods and supervised DNNs in diagnosing gear faults. On the other hand, majority of the available data-driven diagnostic approaches do not consider simultaneous faults in the systems. The proposed diagnostic system is evaluated over different scenarios and settings in diagnosing simultaneous gear faults in electromechanical systems.

4.1 Experimental Setting

The configuration of the electromechanical system under study is illustrated in Fig.4.1. A digital controllable brake is linked to a 250W, 50Hz, 400V, star-connected, 0.77A, 4-pole, 1380rpm, three-phase squirrel-cage induction motor through a one-stage parallel

shaft helical gear with a number of teeth at the input N_{r1}=25 and at the output $N_{r2} = 75$ (Fig. 4.1). f_{r1} and f_{r2} represent the rotation frequencies at the input and the output stages of the gear, respectively. The mechanical load can be modified by adjusting the rotation speed using a digital controllable brake at the output stage of the gear. The system instrumentation consists of three commercial wide-band current sensors with the same 0.1V/A sensitivity with frequency bandwidth of [1Hz, 20MHz]. Moreover, two accelerometers with the sensitivity of 500mV/g and 22kHz frequency bandwidth are installed close to the input and output stages of the gear to measure the mechanical transversal vibration. Also, a torque sensor with 5kHz frequency bandwidth is implemented between the induction machine shaft and the input stage of the gear for torsional vibration analysis. A 24-bit resolution modular data acquisition system with built-in adjustable signal conditioning filters has been used for data collection. The sampling frequency is fixed to $F_s = 5$kHz and the acquisition time is adjusted to $T_{acq} = 60$s for all collected data. Several tests have been carried out on the set-up in both healthy and faulty conditions at five levels of load. For the faulty condition, three groups of tests have been performed including the pinion, the wheel and the simultaneous pinion-wheel gear tooth damage faults in order to make the classification of each particular fault based on the features extracted from the vibration, the measured torque, the SCSVIF and the SCSVIA. In these tests, each faulty pinion and wheel includes only one tooth damaged surface with depth of about 0.3mm in comparison with the healthy gear (Fig. 4.1.b). The last test is related to the healthy condition. In order to collect enough data from the test rig each test has been carried out 10 times.

4.2 Design of the Diagnostic System

The designed hybrid diagnostic framework consists of two major components: (i) feature extraction and (ii) decision making. The former initially processes the captured

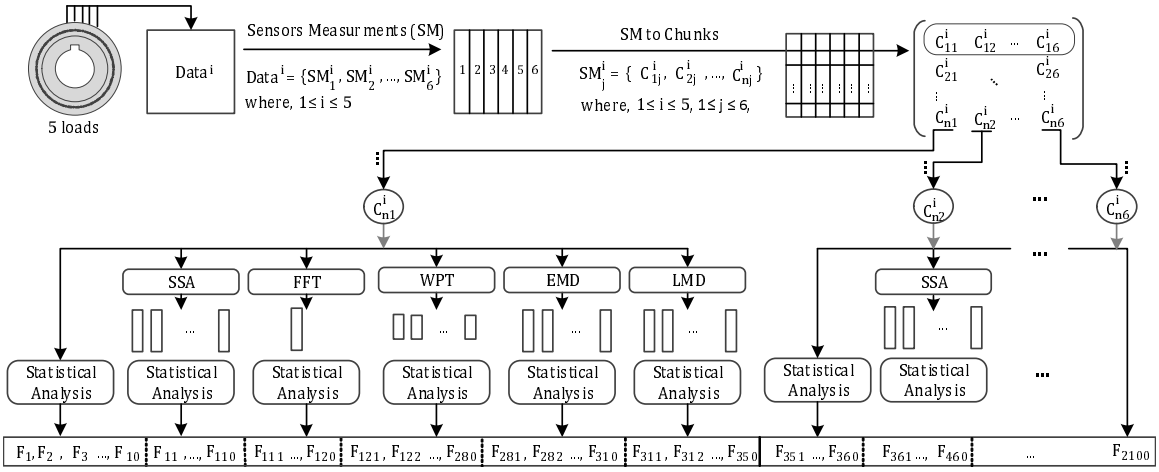

Figure 4.2 – Parallel design of the multi-step feature extraction.

signals through a segmentation phase. Various methods are then utilized to construct a representative pool of features based on the segmented data. The latter enables diagnosing gear faults through SSL and deep learning. The components of this hybrid framework are further explained in the following as they are required for an accurate experimental comparison.

4.2.1 Feature Extraction

For this framework, a multi-step feature extraction is considered to process the raw sensory measurements and extract informative features. Figure 4.2 illustrates the parallel design of the multi-step feature extraction procedure. The raw data are actually several different measurements collected through the data acquisition procedure. These measurements are collected from the three-phase stator currents, the torque sensor (Torque), the vibration sensor at the input stage of gear (ACC1), the vibration sensor at the output stage of gear (ACC2) and the microphone (MIC), that are set into a feature vector as [SCSVIF SCSVIM Torque ACC1 ACC2 MIC]. It should be noted that both SCSVIF and SCSVIM features are determined based on the induction

machine stator current space vector by:

$$i_D(t) = \sqrt{\frac{2}{3}} I_{sA}(t) - \frac{1}{\sqrt{6}} I_{sB}(t) - \frac{1}{\sqrt{6}} I_{sC}(t), \tag{4.1}$$

$$i_Q(t) = \frac{1}{\sqrt{2}} I_{sB}(t) - \frac{1}{\sqrt{2}} I_{sC}(t), \tag{4.2}$$

where i_D and i_Q are the elements of the stator current space vector: $i_D + j \times i_Q$, $j = \sqrt{-1}$, and I_{sA}, I_{sB} and I_{sC} are the stator phase currents. Thus, both SCAVIF and SCSVIA are defined as:

$$SCSVIA = \sqrt{i_D(t)^2 + i_Q(t)^2}, \tag{4.3}$$

$$SCSVIF = \frac{1}{2\pi} \frac{d}{dt} \left(\arctan \left(\frac{i_Q(t)}{i_D(t)} \right) \right). \tag{4.4}$$

The data are collected in healthy and faulty conditions with five different load levels, forming five different datasets $Data^i$, $1 \leq i \leq 5$, as reported in Table 4.1.

Table 4.1 – Load conditions as a function of wheel rotation speed (rpm)

	$Data^1$	$Data^2$	$Data^3$	$Data^4$	$Data^5$
f_{r2}	460	468	472	484	492

In order to process the collected data from the sensors, firstly the segmentation is applied to each feature of the measured dataset. Considering each dataset $Data^i$ contains 6 different sensory measurements (SM), so that $Data^i = \{SM_1^i; SM_2^i; \ldots; SM_6^i\}$. Segmentation provides n non-overlapping chunks $\{C_{11}^i, \ldots, C_{1n}^i\}$ for each sensory measurement. Each chunk of data contains 1000 data samples. By performing each feature extraction technique, in parallel, more informative features are expected to be achieved.

In the next step, each chunk of data that contains sensor measurements is analyzed in three different domains; Time-domain, Frequency-domain and Time-Frequency

domain. To this aim, the advanced methods in these domains (i.e., Fast Fourier Transform (FFT), Singular Spectrum Analysis (SSA)[59], Wavelet Packet Transform (WPT) [55], Empirical Mode Decomposition (EMD) [56] and Local Mean Decomposition (LMD) [57]) are selected and applied on each chunk of data. Then, 10 different statistical measurements are computed over the obtained results of each domain. The brief explanation of each domain is provided as follows:

This helps to reveal the hidden characteristics of each chunk of sensory data. Here, 10 different fault indicators, that are widely used in many literatures are calculated [15, 56]. These statistical measurements include peak to peak, mean value, variance, root mean square, skewness, kurtosis, crest factor, impulse factor, margin factor, and Energy. As it is presented in Figure 4.2, this approach creates 10 new features (F_1 to F_{10}) from each chunk. In parallel, SSA is also applied on the chunk of data [59, 57]. Indeed, SSA decomposes a signal into several numbers of components, while embedding data into a Hankel matrix and applying singular value decomposition (SVD). The highest the number of eigenvalues, the more informative components can be obtained. Here, 10 informative components are initially extracted by SSA, and, consequently, statistical measures of each extracted component are calculated, forming 10×10 new features (F_{11} to F_{110}).

In the view of frequency-domain analysis, the spectrum of each chunk is obtained by applying FFT and, then, statistical characteristics of the frequency spectrum are calculated (F_{111} to F_{120}).

In dual-domain, each chunk is decomposed into a several number of components, where each component provides some information in both time-domain and frequency-domain. The dual-domain methods such as WPT, EMD and LMD could manages the none-stationary behavior of the signal, and, thus, are extensively used in diagnostic applications. In wavelet packet transform (WPT) [55], decomposition is applied to both low pass results (approximations) and high pass results (details). The decomposed components (packs), that are obtained through the decomposition up to four

levels, are then used to attain statistical characteristics (F_{121} to F_{280}).

EMD [56] and LMD [57] are also taken into account. EMD and LMD could decompose a non-linear signal into number of components that are called Intrinsic Mode Functions (IMFs) and Product Functions (PFs), respectively, in a way that the summation of the extracted components with the final residual could reconstruct the given input signal [57, 60]. LMD often needs less iterations than EMD to complete the decomposition task, since it uses iterative local mean calculations by means of moving average filter, while EMD needs to recursively perform cubic spline interpolations, as the core of its algorithm. Therefore, LMD has been applied as a suitable decomposition method for real-time system in some literature [61, 57, 60].

Upon decomposition of each chunk by means of EMD and LMD, correspondingly, 10 statistical measurements are computed resulting in 30 features (F_{281} to F_{310}) and 40 features (F_{311} to F_{350}), respectively. Since, each raw sensory dataset contains 6 different features (i.e., SCSVIF, SCSVIM, Torque, ACC1, ACC2, MIC), the multi-step feature extraction results in a pool of $350 \times 6 = 2100$ informative features.

4.2.2 Decision Making

Semi-supervised learning methods can handle the problem of classification in the presence of a large volume of unlabeled samples, while only a few labeled samples are available. Here, we aim to study that considering the rarity of the labeled samples in SSL approaches, to what extent semi-supervised deep learning can be beneficial.

This component adapts a powerful algorithm for semi-supervised deep learning, so-called Semi-Supervised Deep Ladder Network (SSDLN), for decision making and diagnosing gear faults. In addition, several advanced supervised and semi-supervised approaches have been devised in the decision making component of the diagnostic system along with SSDLN. This enables a comparative study between SSDLN and not only advanced SSL methods, but also other advanced supervised DNNs. This also reveals the advantages of SSDLN over conventional approaches in SL and SSL.

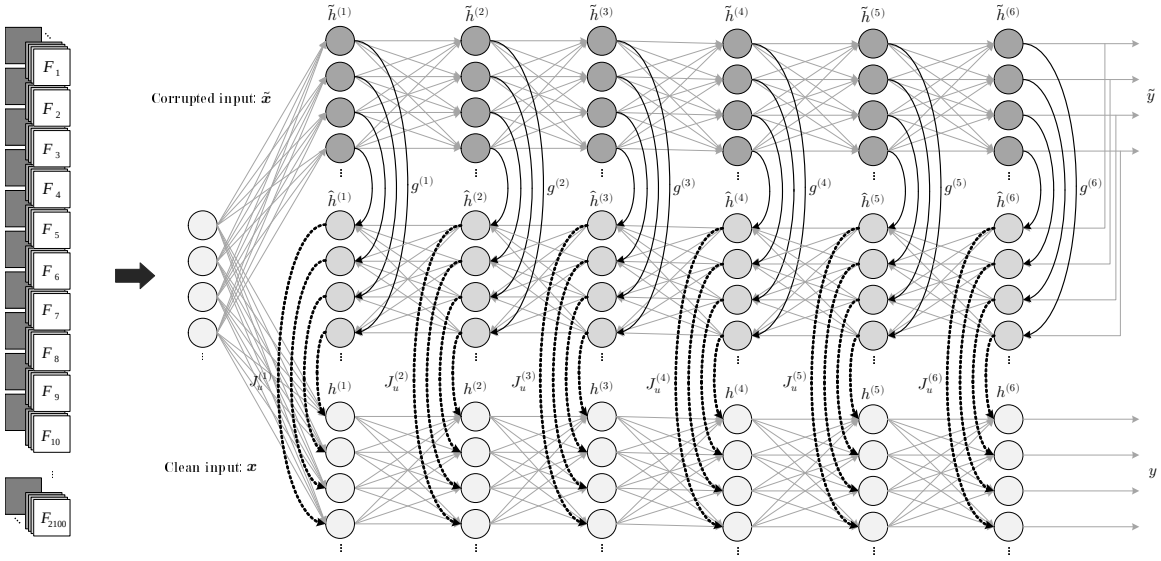

Figure 4.3 – The illustration of a semi-supervised ladder network with the number of layers $\Gamma = 6$. $g^{(\iota)}$ and $J^{(\iota)}$ are shown by solid and dotted vertical curved lines, respectively. Neurons with corrupted, denoised, and clean input are specified with dark gray, light gray, and white circles, respectively.

SSDLN is briefly explained in the following.

Semi-Supervised Deep Ladder Network

The learning procedure of SSDLN [58] is accomplished through the integration of the supervised and unsupervised learning (UL) strategies. This has been done by modeling hierarchical latent variables $\{h^{(\iota)}\}_{\iota=1}^{\Gamma}$, where Γ is the number of layers in the network. The procedure is generally divided into inference and learning tasks. The former is the process of estimating the posterior probability of an unseen h, which is undertook by computing a denoising function between the distorted data $\tilde{x}$ and the original source x, as illustrated in Fig. 4.3. The latter, on the other hand, is performed by minimizing the denoising cost function $J_u^{(\iota)}$ between the estimated $\hat{h}^{(\iota)}$ and the clean latent variables $h^{(\iota)}$ in the ι-th layer of the network. $\hat{h}^{(\iota)}$ is calculated through a denoising function $g^{(\iota)}$ as follows:

$$\hat{h}^{(\iota)} = g(\tilde{h}^{(\iota)}, u^{(\iota)}) = \Big(\tilde{h}^{(\iota)} - \eta(u^{(\iota)})\Big)\varphi(u^{(\iota)}) + \eta(u^{(\iota)}), \tag{4.5}$$

where $\tilde{h}$ is the noisy input, and assuming that $\Pi^{(\iota)}$ is a matrix containing weights from layer $\iota + 1$ to ι, $\sigma(.)$ is the standard deviation function, and $\mu(.)$ is the mean function, $u^{(\iota)}$ is the information propagated from higher layers through the backward path:

$$u^{(\iota)} = \frac{\Pi^{(\iota)}.\hat{h}^{(\iota+1)} - \mu(\Pi^{(\iota)}.\hat{h}^{(\iota+1)})}{\sigma(\Pi^{(\iota)}.\hat{h}^{(\iota+1)})}, \tag{4.6}$$

and functions η and φ apply expressive nonlinearities:

$$\eta(u^{(\iota)}) = a_1^{(\iota)}\mathcal{S}(a_2^{(\iota)}u^{(\iota)} + a_3^{(\iota)}) + a_4^{(\iota)}u^{(\iota)} + a_5^{(\iota)}, \tag{4.7}$$

$$\varphi(u^{(\iota)}) = a_6^{(\iota)}\mathcal{S}(a_7^{(\iota)}u^{(\iota)} + a_8^{(\iota)}) + a_9^{(\iota)}u^{(\iota)} + a_{10}^{(\iota)}, \tag{4.8}$$

where $a_1, a_2, ..., a_{10}$ are unit-wise parameters of the encoder, and $\mathcal{S}$ stands for the sigmoid function, $\mathcal{S} = \frac{1}{1+e^{-1}}$.

Supervised and unsupervised components of SSDLN work at the same time, as shown in Fig. 4.3, in order to minimize an objective function, which is equal to the sum of two cost functions. Given a training set with n_l labeled samples $(x_{(1)}, y_{(1)}), (x_{(2)}, y_{(2)}), ..., (x_{(n_l)}, y_{(n_l)})$, and n_u unlabeled samples $x_{(1)}, x_{(2)}, ..., x_{(n_u)}$, the objective function J is then reformulated as follows:

$$\begin{aligned} J = -\sum_{i=1}^{n_l} \log P(\tilde{y}_{(i)} = y_{(i)}|x_{(i)}) + \\ \sum_{i=1+n_l}^{n_u}\sum_{\iota=1}^{\Gamma} \lambda_\iota J_u(h_{(i)}^{(\iota)} - \hat{h}_{(i)}^{(\iota)}), \end{aligned} \tag{4.9}$$

where $\tilde{y}$ is the noisy output (see Fig. 4.3), and λ_ι stands for the denoising cost multiplier, which is tuned as a hyperparameter. In Equation (4.9), the first term stands for the cross entropy of the distorted output which resembles the supervised

cost, while the second term penalizes the unsupervised denoising cost at each layer. The denoising cost is computed as follows:

$$J_u(h^{(\iota)}, \hat{h}^{(\iota)}) = \left\| \frac{\hat{h}^{(\iota)} - \mu(\Upsilon^{(\iota)}.\tilde{\upsilon}^{(\iota-1)})}{\sigma(\Upsilon^{(\iota)}.\tilde{\upsilon}^{(\iota-1)})} - h^{(\iota)} \right\|_2^2, \tag{4.10}$$

where $\Upsilon^{(\iota)}$ and $\upsilon^{(\iota-1)}$ stand for the matrix of weights from layer $\iota - 1$ to ι and the post-activation at layer $\iota - 1$, respectively.

The main feature of SSDLN is the skip connections (see curved connections in Fig. 4.3) connecting encoders and decoders in each layer of the network [62]. This relaxes the data representation in the higher layers as the omitted details can be fetched by the decoder in each layer. Furthermore, in order to compute the denoising cost function, the encoder is trained for both the clean and the noisy latent variables in a parallel scheme.

Semi-Supervised Classifiers

Two SSL classifiers considered for comparison including graph-based algorithms namely Semi-supervised Extreme Learning Machine (SSELM) [42], and Low Density Separation (LDS) [41], which are explained in Chapter 3.

Supervised Classifiers

To compare the performance of the SSDLN with advanced SL algorithms, two of the most succesful deep learners such as the Convolutional Neural Network (CNN) [63] and the Long Short-Term Memory network (LSTM) [64] are used in this study for the sake of comparison:

1) Convolutional Neural Network: The CNN structure consists of multiple hidden layers that can be categorized into three types: convolutional, pooling, and fully connected layers. The first type contains filtered features, which are attained by processing different parts of the features. The pooling layers normalize and shrink

the output of the previous layer. The combination of these two layers can be stacked, and, then, the output of the last layer goes through a fully connected layer, resulting in the CNN output [63].

2) Long Short-Term Memory Network: LSTM is an advanced type of Recurrent Neural Network (RNN) [65], which benefits from an intrinsic memory (or cell), in which predictions are memorized over time to perform a more accurate decision making w.r.t. the history of its predictions in the current time-step and those occurring in time.

4.3 Experimental Results

Here, the experimental setting is initially explained, and, then, the achieved results are presented and compared.

To evaluate the designed diagnostic framework, five different experiments have been considered. Each experiment is repeated ten times and the results are then averaged. In this procedure, the processed data for each experiment is divided into train and test subsets with a 1:4 ratio, where only 100 samples are labeled in the former (roughly 1% of the train set), and the rest remain unlabeled. These sets contain equal number of samples for each class.

Deep learning is performed through 500 epochs. Different values for the initial learning rate have been tried between 0.001 and 0.02, and based on these trials, it is set to 0.01. Parameter optimization for each method is performed by means of Adam optimizer [66].

4.3.1 Experimental Comparison

The attained results are analyzed in terms of accuracy and standard deviation.

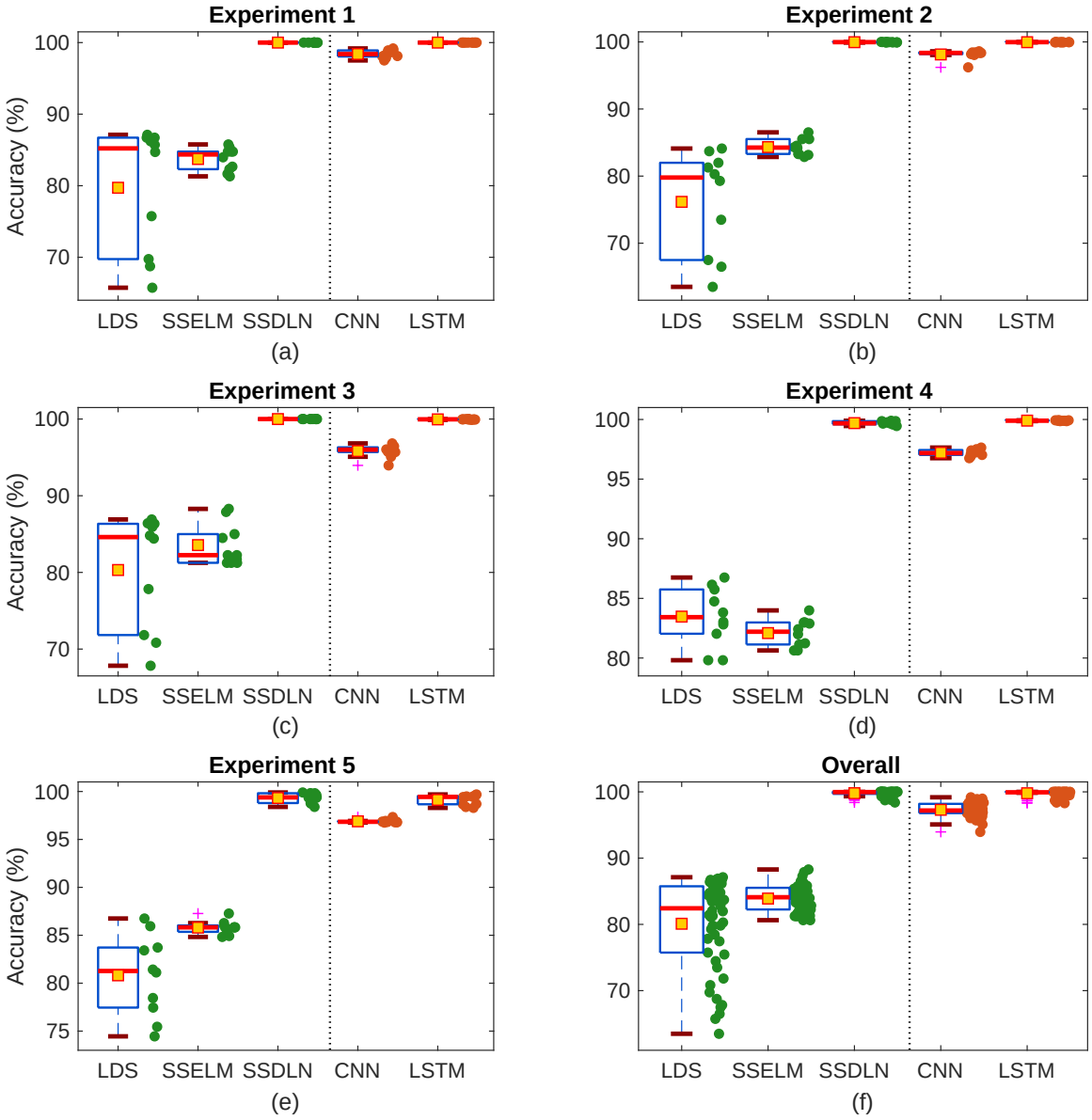

Figure 4.4 – The distribution of the attained accuracies for each experiment is illustrated in (a-e). The averaged results over all the experiments are represented in (f). Mean, accuracy, and outlier are shown with solid square, circle, and plus sign, respectively. Upper bound, lower bound and median are specified by horizontal lines located on top, bottom and inside of each box. The left-hand side and right-hand side of the dotted line show the results for SSL and SL algorithms, respectively.

Accuracy

Figures. 5.3(a-e) show the achieved accuracies for each experiment. The attained accuracies over all experiments and performed runs are also illustrated in Fig. 5.3(f).

Table 4.2 – The averaged accuracies ± standard deviations of all runs for each experiment over ten trials.

Algorithm	Accuracy (%)					
	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	#
LDS	79.716 ± 8.734	76.161 ± 7.770	80.311 ± 7.524	83.472 ± 2.455	80.819 ± 4.260	5
SSELM	83.734 ± 1.613	84.318 ± 1.218	83.561 ± 2.725	82.090 ± 1.148	85.798 ± 0.694	4
SSLDN	**100.000** ± 0.00	**99.958** ± 0.034	**100.000** ± 0.00	99.695 ± 0.146	**99.323** ± 0.5204	1
CNN	98.401 ± 0.541	98.131 ± 0.691	95.826 ± 0.810	97.213 ± 0.284	96.888 ± 0.166	3
LSTM	**100.000** ± 0.00	99.957 ± 0.018	99.959 ± 0.031	**99.904** ± 0.026	99.124 ± 0.546	2

The averaged values of different runs for each experiment are listed in Table 4.2, in which the bold entries show the maximum accuracy achieved for each experiment. Considering all the experiments, SSDLN, LSTM and CNN are ranked from first to third, respectively. SSELM and LDS are ranked as fourth and fifth ranks subsequently. This reveals the strength of deep learning in extracting useful features from the input data.

Generally, SSL algorithms are expected to be more efficient, but less accurate than SL, due to the rarity of the labeled samples in SSL algorithms. This can be seen by comparing the results of LDS and SSELM with CNN and LSTM. Nonetheless, the superior accuracy of SSDLN shows that not only it copes with the rarity of the labeled samples, but also outperforms SL algorithms.

Standard Deviation

The reliability of these classification algorithms is then evaluated with respect to the standard deviation of the attained results over various runs. In Fig 5.3, the length of the bars indicate the averaged accuracy and standard deviation of each algorithm over five experiments and ten different runs. In addition, the standard deviations of the attained accuracies by each algorithm over all runs for each experiment are listed in Table 4.2.

Among the deep learners, SSDLN, LSTM, and CNN are ranked from first to

third in terms of stability. The maximum value of the standard deviation at each experiment for SSDLN and LSTM is about 0.5%, which indicates their stability.

Due to the rarity of the labeled samples in SSL, semi-supervised learners should make predictions based on the available unlabeled samples in the training set. Such predictions may not be identical to the true labels, which lower the accuracy of the constructed classification model. However, under the utilized setting, the SSL component of SSDLN has made it even more stable than the supervised DNNs, i.e., CNN and LSTM.

4.3.2 Discussion

The attained results in such a harsh learning condition show that SSDLN is the most stable method and diagnoses gear faults with the higher accuracy compared to other algorithms.

Fig. 4.5 shows the overall performance for each method through all the experiments and runs. SSDLN achieves the maximum accuracy and the lowest standard deviation. This figure shows SSDLN outperforms other classifiers. This is while LSTM as a supervised method is ranked second. This indicates that although LSTM receives all labels in a supervised setting, it still slightly underperforms SSDLN. Fig. 4.5 also shows that after SSDLN, all SL-based algorithms including LSTM and CNN outperform SSL-based methods, i.e., LDS and SSELM. This is due to the fact that these SL-based methods are implemented in a softer learning condition, where all samples have labels during the training session, and, thus, it is expected for them to achieve a higher classification accuracy.

Another advantage of SSDLN compared to other deep learners is its fast convergence. Fig. 4.6 illustrates the error rate of the deep learners during their training process. SSDLN rapidly converges and builds a precise model. On the other hand, LSTM requires a lot of epochs to reach a proper classification model. CNN converges at a slower rate compared to SSDLN; however, it is much faster than LSTM. Although

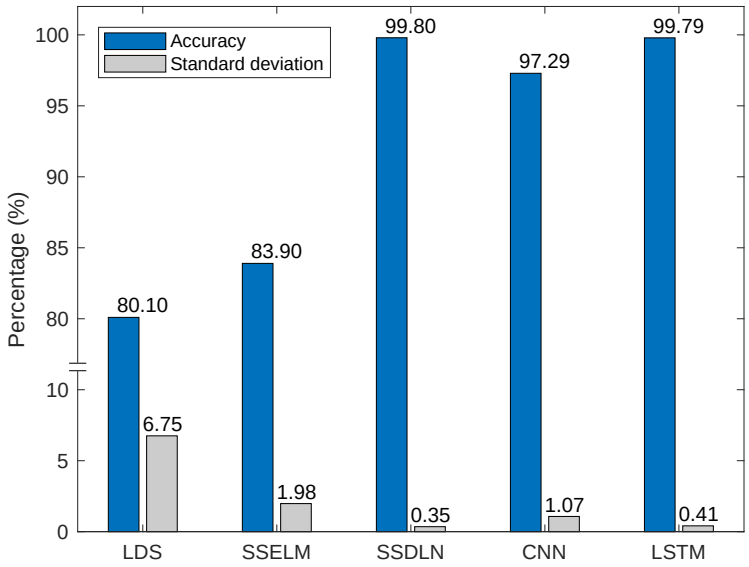

Figure 4.5 – The averaged accuracies and standard deviations of each algorithm over five experiments and ten runs.

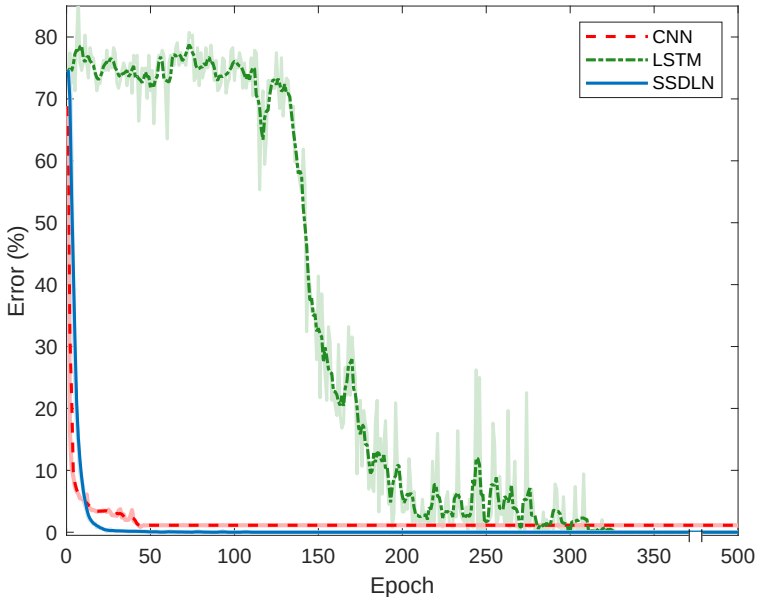

Figure 4.6 – The smoothed test error achieved through the training process of each deep learner in the third scenario.

here the number of epochs is set to 500, such a large value has been merely set to enable a more detailed study of the utilized algorithms. This is while SSDLN can use smaller values for the number of epochs and still maintains a superior performance.

4.4 Summery

In this chapter, a hybrid framework is proposed for diagnosing simultaneous gear faults in electromechanical systems. A semi-supervised deep learning algorithm has been adapted and devised in this hybrid framework to facilitate decision making in a very common and realistic condition. This is a harsh learning condition of the high-dimensional feature space and the rarity of the labeled samples along with the excess of the unlabeled samples in the input space. The fault diagnosis has been performed by means of feature extraction and decision making components. In the former, several advanced approaches, in parallel, extract informative features from the raw signals collected from the gearbox and form a representative high-dimensional feature space. In the latter, then, the adapted semi-supervised deep learner, SSDLN, learns from this high-dimensional feature space in the presence of rarely labeled samples, and diagnoses the gear faults. In addition, advanced DNNs are adapted for fault classification in order to enable a comparative study. The achieved results show the superiority of SSDLN compared to other methods including SL-based DNNs that are implemented in a much softer learning condition, where all samples were labeled during the training session. This conclusion is very important, since it contradicts the general expectation as the supervised learning usually solves a less restrictive problem compared to the semi-supervised learning.

Chapter 5

Fault Diagnosis with Limited Supervision in Non-stationary Environments

Data-driven FDS approaches are usually trained based on collected data in stationary environments (i.e., data are sampled from an independent and identically distributed (i.i.d.) process). On the other hand, collecting representative data is often a dynamic process of successive data acquisition campaigns. In such non-stationary environments, data patterns become available successively, over a period of time. Therefore, designing efficient data-driven schemes are typically preferred, since these techniques are more robust against arbitrary data distributions and types of faults [11]. Furthermore, they do not rely on seasonal changes in patterns (i.e., a stationarity assumption). Our setting assumes that data arrive in a non-stationary environment, where sensor data collection forms a data stream [12].

In case of a non-stationary data stream, the model used for the FDS must be adaptive to adapt and react to the changes in the stream. Not incorporating an adaptation mechanism can result in a model that becomes obsolete quite quickly depending on the rate of change. Although several methods have been proposed to handle concept drift using both techniques from SL and SSL [67, 68], they are still limited in some senses. Algorithms such as Learn^{++}.NSE [69] are limited by SL, and passive learning used in most of SSL algorithms, such as Weight Estimation Algorithm [67], can miss the opportunity to exploit that an oracle could be available to provide labels based on an available budget.

Diagnosing faults in non-stationary environments is a challenging task. On the other hand, providing FDS with external updates is not feasible, since labeling the

unlabeled data is an expensive and offline task. To overcome these issues, recent works have been developed to address Extreme Verification Latency (EVL), which is when the labeled data is merely available at the first time step. A more realistic learning setting for the design of FDS aligns better with EVL. Although EVL learners can be used for diagnosing faults from the data stream, they have some restricting assumptions. EVL assumes that the labeled data is only provided at the initial time step, and the subsequent samples are fully unlabeled, which is a more realistic learning scenario for FDS. EVL classifiers (EVLC) are mostly limited to work with gradual drifts and the fixed number of classes, while there is a need to work with both abrupt and gradual changes in the data stream, and, moreover, handle new classes in the subsequent instalments, without any external updates.

In this chapter, a framework for fault detection and classification that is accomplished on the data stream with both gradual and abrupt drifts. The framework is only provided with prior information about possible faults at the initial step; however, despite this the framework can still detect novel faults without receiving any update. Furthermore, an efficient fault classification algorithm is presented to maximize the efficiency of the proposed framework. Finally, the proposed framework is applied for diagnosing bearing defects in induction motors to demonstrate its feasibility for industrial applications.

5.1 Related Work

EVL is a challenging problem that has recently received more attention in the computational intelligence community. The arbitrary subpopulation tracker (APT) [70] algorithm is one such approach for EVL. APT works by using a two-step procedure: (1) a one-to-one assignment from unlabeled to labeled data is performed using Expectation Maximization (EM); then, (2) the classifier is updated to reflect the current sub-population parameters. APT requires that all the sub-population parameters be

made available at the time of initialization and the nonstationarity in the data be structured or systematic, which may not be the situation with $\mathcal{CD}_{\mathcal{A}}$. Unfortunately, there is a significant computational overhead with APT.

Stream Classification Algorithm Guided by Clustering (SCARGC) [71] is another EVL approach that focuses on computational efficiency. SCARGC performs clustering of the data that is followed by classification. Similar of this process is repeated to use the unlabeled data to track the non-stationary environment.

COMPact Object Sample Extraction (COMPOSE) [14, 72, 73] is an EVLC that aims to classify the unlabeled data using a SSL base learner with the labeled data at the initial step then extract core supports from the classified samples to retrain the SSL base classifier. Core supports could be extracted through shrinkage of an α-Shape [14], or by fitting a Gaussian mixture model (GMM) to the data [72]. FAST COMPOSE [73], on the other hand, uses all the classified samples instead of core support extraction. COMPOSE was empirically shown to perform better than APT, and Fast COMPOSE significantly improves the runtime over COMPOSE.

5.2 Design of the Diagnostic System

Despite all advantages of the EVLCs, they are not able to handle all types of drifts. The proposed framework, called drift and novelty class detection and adaptation under extreme verification latency (DISCOVERY), (see Fig. 5.1) provides more flexibility for EVLC than traditional FDS approaches by incorporating an adaptation mechanism upon the occurrence of an abrupt drift $\mathcal{CD}_A$. The proposed framework accomplishes this by generating internal updates whenever $\mathcal{CD}_{\mathcal{A}}$ takes place. These updates contain novel information about the presence of unknown classes and the abrupt change of the known distribution.

Our framework consists of two major modules: (1) a two-stage detector, and (2) a fault classifier. The former contains an abrupt drift detector and a new class detector.

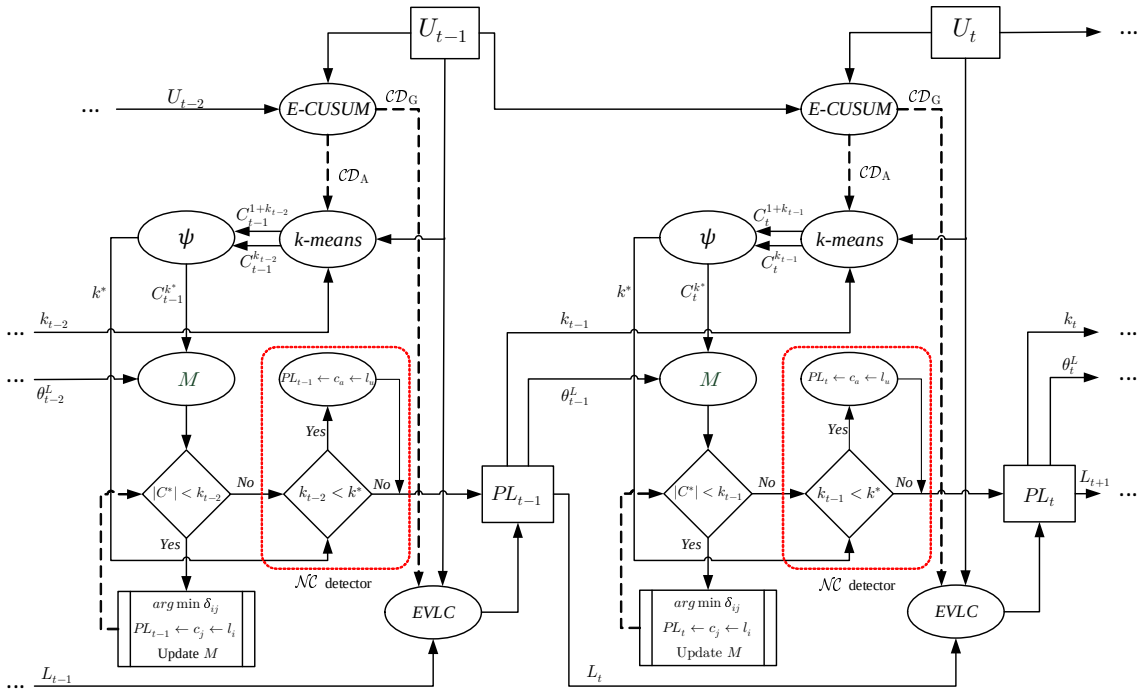

Figure 5.1 – The block diagram of the DISCOVERY diagnostic system. The proposed scheme contains a two-stage detector including E-CUSUM and $\mathcal{NC}$ detector, and an EVLC module which can be replaced by SCARGC or any variations of COMPOSE including the novel Affinity-based COMPOSE. [2]

The latter is the proposed EVLC named as Affinity-based COMPOSE.

5.2.1 Two-stage Drift Detector

To make the framework robust against any forms of drift, the first yet most important step is to detect the type of drift at each time step t, where $t = \{1, ..., \mathcal{T}\}$ (steps 1-12 in Algorithm 3). Here, this has been done by resorting to the E-CUSUM algorithm [74]. The E-CUSUM module in Fig. 5.1 is explained in the following.

Abrupt Drift Detector

E-CUSUM aims to model the changes in the data sequence Q (i.e., in Fig. 5.1, Q is the union of the previous batch and the current batch $Q = U_{t-2} \cup U_{t-1}$) by different parametrizations (Θ^0 and Θ^1) of a probability density function P so that a drift can

be detected by comparing the change ratio with a threshold τ.

By leveraging the central limit theorem, the base parameters $\Theta_0 = \{\hat{\mu}_0, \hat{\sigma}_0^2\}$ are estimated by computing the mean $\hat{\mu}$ and the standard deviation $\hat{\sigma}^2$ on a transformed sequence Q_T:

$$Q_T = \bigcup_{i=1}^{\frac{m}{n}} s(i) : \ s(i) = \sum_{h=1+n(i-1)}^{ni} x_h/n \tag{5.1}$$

where x_h is a sample, and m is the number of samples in Q; and n stands for a number satisfying the theorem. The alternative hypothesis $\mathcal{H}_1 : \Theta^1 = \{\hat{\mu_1}, \hat{\sigma_1}^2\}$ is defined as being outside of the P_{Θ^0} domain. Given a confidence parameter ω, an interval of $(1-\omega)100\%$ is defined on Θ^0, i.e., $[\hat{\mu}_{min}, \hat{\mu}_{max}]$ and $[\hat{\sigma}^2_{min}, \hat{\sigma}^2_{max}]$, to facilitate the estimation of Θ^1. Then, $\hat{\mu_1} = [\hat{\mu_1}_{min}, \hat{\mu_1}_{max}]$ and $\hat{\sigma_1}^2 = [\hat{\sigma_1}^2_{min}, \hat{\sigma_1}^2_{max}]$ in Θ^1 are calculated as follows:

$$\hat{\mu}_1 = \hat{\mu}_0 \pm \lambda \sqrt{\frac{n}{m}} \varphi_{\omega/2} \hat{\sigma}_0 \tag{5.2}$$

$$\begin{cases} \hat{\sigma}^2_{1_{max}} = \hat{\sigma}_0^2 + \lambda(\hat{\sigma}^2_{max} - \hat{\sigma}_0^2) \\ \hat{\sigma}^2_{1_{min}} = \hat{\sigma}_0^2 + \lambda(\hat{\sigma}_0^2 - \hat{\sigma}^2_{min}) \end{cases} \tag{5.3}$$

where φ and λ are the normal distribution and the sensitivity parameter (i.e., decreasing/increasing results in detecting smaller/bigger changes), respectively. The change ratio R then can be computed as follows:

$$R_i = \sum_{h=1}^{i} \ln \frac{P_{\Theta^0}(x_h)}{P_{\Theta^1}(x_h)}, \quad i = 1, ..., m/n, \tag{5.4}$$

and a degree of change ϵ can be obtained as $\epsilon_i = R_i - \min_{1 \leq h \leq i}(R_h)$. An abrupt drift $\mathcal{CD}_{\mathcal{A}}$ is then detected when $\epsilon_i > \tau$, where $\tau = \max_{1 \leq i \leq m/n}(\epsilon_i)$ is the threshold.

Given an input signal of unlabeled data $U_t = \{x_1^t, ..., x_{n_u}^t\}$ at time step t (see Fig. 5.1), the algorithm merges U_t with the collected data at the previous time step $t-1$, U_{t-1}, and forms $Q = \{U_{t-1} \cup U_t\}$, and, then, feeds it to the detector. The detector checks all features of Q one by one and transforms them into Q_T, and, then, activates

an alarm upon occurrence of an abrupt change $\mathcal{CD}_{\mathcal{A}}$.

New Class ($\mathcal{NC}$) Detector

As illustrated in Fig. 5.1, if $\mathcal{CD}_{\mathcal{A}}$ is detected (step 14 in Algorithm 3) then the adaptation procedure begins. The goal of this phase is to reconstruct the hypothesis space for the EVLC so that the online monitoring can be continued without the need for manual update. To do so, first the root cause of the drift should be specified (steps 15-18 in Algorithm 3). A $\mathcal{CD}_{\mathcal{A}}$ occurs as a result of a new class or an abrupt change of the seen distributions. To detect the root cause of $\mathcal{CD}_{\mathcal{A}}$, U_t is clustered using the k-means module (see Fig. 5.1) for k_{t-1} and $1 + k_{t-1}$ number of clusters. Note that the number of classes in the initial time step is known so initially k_0 is equal to the number of classes. The attained partitions for each run of k-means clustering, i.e., clustering outputs $C_t^{k_{t-1}} = \{c_i\}_{i=1}^{k_{t-1}}$ and $C_t^{1+k_{t-1}} = \{c_i\}_{i=1}^{1+k_{t-1}}$ are then fed to a Silhouette function (ψ module in Fig. 5.1) to find out which k results in a more accurate clustering [75]. The Silhouette function ψ measures the partition quality by indicating how well x_i^t is assigned to the clusters, where x_i^t is the i-th sample in the received batch at time-step t. To do this, ψ gives a coefficient value that is computed for x_i^t by means of the between and within cluster dissimilarity, ϱ_b and ϱ_w as follows:

$$\psi(x_i^t) = \begin{cases} 1 - \varrho_w(x_i^t)/\varrho_b(x_i^t), & \varrho_w(x_i^t) < \varrho_b(x_i^t) \\ 0, & \varrho_w(x_i^t) = \varrho_b(x_i^t) \\ \varrho_b(x_i^t)/\varrho_w(x_i^t) - 1, & \varrho_w(x_i^t) > \varrho_b(x_i^t) \end{cases} \tag{5.5}$$

Our algorithm then compares the sum of ψ coefficient for all samples $\sum_{i=1}^{n_u} \psi(x_i^t)$ attained by each partitioning $C_t^{k_{t-1}}$ and $C_t^{1+k_{t-1}}$, and consequently it returns the cluster which resulted in the larger sum of ψ coefficients as a proper partition and assigns the k of that partition as the true number of clusters, k^*. We consider the addition of one new class in the designed experiments. However, the proposed framework can

detect the presence of multiple new classes at each time step in an iterative procedure, where the algorithm iteratively increases the number of partitions k until the sum of coefficients ψ has been decreased. This can be done by resorting to the k-means and ψ modules as shown in Fig. 5.1.

Once the true number of clusters k^* has been assigned, the algorithm can assign the label of gradually drifted classes to them (steps 19-25 in Algorithm 3). Then the presence of a new class can be detected through the $\mathcal{NC}$ detector in Fig. 5.1 (steps 26-28 in Algorithm 3). To do so, the set of centroids for the current time step $\theta_t^U = \{\mu_j\}_{j=1}^{k^*}$ is compared with the labeled centroids of the previous time step $\theta_{t-1}^L = \{\mu_i\}_{i=1}^{k_{t-1}}$. A distance matrix $M = \{\delta_{ij}\}$ is then constructed for comparison, where $\delta_{ij} = \|\mu_i - \mu_j\|$. The M module in Fig. 5.1 refers to this step. We then find the closest pair of centroids $\{\forall\ 1 \leq j \leq k^*, 1 \leq i \leq k_{t-1} | \operatorname{argmin} \delta_{ij}\}$, where all samples of $c_j \in C^{K^*}$ are assigned to the class label l_i of the paired cluster $c_i \in C^{k_{t-1}}$ and $l_i \in \Omega = \{l_i\}_{i=1}^{k_{t-1}}$ is a set of distinct class labels. Then we update M by replacing all elements of the i-th row and the j-th column with a fixed value $\max(\delta_{ij}) + \zeta$, where ζ is a small number. This subroutine is shown by a rectangular box at the bottom of Fig. 5.1. This avoids assigning multiple distinct labels to the samples of the same clusters. It then iteratively stores the newly labeled clusters in C^*, which was initially set to $\varnothing$ (see the closed loop between the condition $|C^*| < k_{t-2}$ and the rectangular subroutine under the M module in Fig. 5.1). This iterative process stops when $|C^*| = k_{t-1}$. Thus far, samples of previously seen classes that are abruptly drifted in the feature space are properly labeled. Otherwise, if $k_{t-1} < k^*$, it means that $\mathcal{CD}_{\mathcal{A}}$ is caused by the presence of new unseen classes, and, thus, one or more centroid in C^{k^*} are still left unpaired to a class. Such centroids reveal the presence of a cluster of samples c_a that belongs to a new unseen class. Afterwards, it assigns an unknown label l_u to the samples of the unpaired cluster c_a. The $\mathcal{NC}$ detector is illustrated with a dotted box in Fig. 5.1. Pseudo-labels are then assigned to all unlabeled samples of U_t and a set of labeled samples $PL_t = \{x_i^t, y_i^t\}_{i=1}^{n_u}$ is formed,

Algorithm 3: DISCOVERY (FDS framework)

Definitions: ψ is the Silhouette function, M is the distance matrix of centroids, λ is the sensitivity parameter of E-CUSUM

Input: U_t, $\mathcal{T}$ #. of time steps, λ, EVLC

Initialisation: Classify the first batch ($t = 1$) of the unlabeled samples U_1 using the provided labeled set L_1 and form: $[PL_1] = \text{EVLC}(U_1, L_1)$

for $t = 2, ..., \mathcal{T}$ **do**
  for each feature of $Q = \{U_{t-1} \cup U_t\}$ **do**
    Compute $\Theta^0 = \{\mu_0, \sigma_0^2\}$ on Q_T using (5.1)
    Compute $\Theta^1 = \{\mu_1, \sigma_1^2\}$ using (5.2) and (5.3)
    for $\forall x \in \{U_{t-1} \cup U_t\}$ **do**
      Calculate R and ϵ using (5.4)
      Estimate τ
      if $\epsilon_i > \tau \vee \epsilon_i < -\tau$ **then**
        Detect $\mathcal{CD}_A$ and activate an alarm
      end if
    end for
  end for
  Collect previously labeled samples PL_{t-1}
  if $\mathcal{CD}_A$ is detected **then**
    $k_{t-1} = |\Omega_{t-1}|$, where $\Omega_{t-1} = \lceil Y_{t-1} \rceil$, $Y_{t-1} \subset PL_{t-1}$. $|.|$ stands for cardinality, and $\lceil . \rceil$ returns unique elements
    $C_t^{k_{t-1}} = k\text{-means}(U_t\ , k_{t-1})$
    $C_t^{1+k_{t-1}} = k\text{-means}(U_t\ , 1 + k_{t-1})$
    Compare $\psi(C_t^{k_{t-1}})$ and $\psi(C_t^{1+k_{t-1}})$ using (5.5) to determine the best number of clusters k^*
    $\forall c_j \in C_t^{k^*}$: $\theta_t^U = \{\mu_j\}_{j=1}^{k^*}$, where $\mu_j = \text{mean}(c_j)$
    $\forall c_i \in C_{t-1}^{k_{t-1}} \leftarrow PL_{t-1}$: $\theta_{t-1}^L = \{\mu_i\}_{i=1}^{k_{t-1}}$, where $\mu_i = \text{mean}(c_i)$
    $\forall \mu \in \{\theta_t^U \cup \theta_{t-1}^L\}$: $\delta_{ij} = ||\mu_i - \mu_j||$, and

$$M = \begin{bmatrix} \delta_{11} & \cdots & \delta_{1k^*} \\ \vdots & \ddots & \vdots \\ \delta_{k_{t-1}1} & \cdots & \delta_{k_{t-1}k^*} \end{bmatrix}$$

    for $i = 1, ..., k_{t-1}$ **do**
      Assign l of $\mu_i \in \theta_{t-1}^L$ to $c_j \in C_t^{k^*}$ corresponding to $\arg\min_{1 \le j \le k^*}(\delta_{ij})$
      Replace i-th row and j-th column of M with $\max(\delta_{ij}) + \zeta$, where ζ is a small number
    end for
  if $k_{t-1} < k^*$ **then**
    Assign an unknown class l_u to $c_a \in C_t^{k^*}$
  end if
  Form a set of pseudolabeled samples: $PL_t \leftarrow C_t^{k^*}$ **else**
    Call EVLC module: $[PL_t] = \text{EVLC}(U_t, L_t)$, where $L_t = PL_{t-1}$
  end if
end for

where $y_i \in Y = \{y_i | y_i \in \Omega\}_{i=1}^{n_u}$. This pseudo-labeled set can be used as a prior knowledge for the upcoming iteration $t+1$ to reconstruct the hypothesis space of the EVLC.

5.2.2 Extreme Verification Latency Classifier Module

An EVLC module (see Fig. 5.1) is designed to address gradual concept drift $\mathcal{CD}_{\mathcal{G}}$. If $\mathcal{CD}_{\mathcal{G}}$ is detected then the proposed framework leverages the EVLC to classify the unlabeled set U_t (step 31 in Algorithm 3).

An EVLC receives a small set of labeled samples $L_t = \{x_i^t, y_i^t\}_{i=1}^{n_l}$ if $t = 1$, which is the initial time step, or the set of pseudo-labeled samples is received if $t > 1$. The pseudo-labels are generated in the previous step by means of EVLC (if $\mathcal{CD}_{\mathcal{G}}$), or resulted through the adaptation phase (if $\mathcal{CD}_{\mathcal{A}}$) as shown in Fig. 5.1. In this study, SCARGC and COMPOSE framework are adapted in the EVLC module. COMPOSE has less limiting hypothesis compared to SCARGC and, thus, provides the framework with more flexibility.

Variants of the COMPOSE have been used as EVLC in the proposed framework. These variations include α-Shape, Gaussian Mixture Models (GMM), and FAST COMPOSE. Among these variations, FAST COMPOSE can be the best choice for online diagnostic applications due to its low computational burden. Nevertheless, it is shown this low computational burden is resulted by sacrificing a slight amount of accuracy as it removes the core-support extraction phase, and instead, uses all available pseudo-labeled samples [73].

Affinity-based COMPOSE

Accuracy is a very important factor for the diagnostic system as is the runtime of the algorithm. To compromise between the accuracy and the runtime, in this paper, we propose Affinity-based COMPOSE, which is shown in Algorithm 4. The main

Algorithm 4: Affinity-based COMPOSE (EVLC module)

Definitions:
L_t^* is the set of sampled labeled data from L_t
ρ is the radial basis function kernel
$\hat{Y}_t$ is the set of predicted labels at t
Input: f Semi-supervised learner, ς sampling rate, U_t, L_t

1 Collect unlabeled data U_t
2 **if** $t = 1$ **then**
3 | Set L_1^* equal to provided labeled data: $L_1^* = L_1$
4 **else**
5 | **if** update is available by user **then**
6 | | Collect the labeled data: $L_t^* = L_t$
7 | | **go to** step 14
8 | **else**
9 | | Collect labeled samples L_t
10 | **end if**
11 | Construct the affinity matrix A between L_t and U_t:

$$A = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1n_u} \\ \vdots & \ddots & \vdots \\ \rho_{n_l 1} & \cdots & \rho_{n_l n_u} \end{bmatrix},$$

$$\rho_{ij} = \exp\Big(\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\Big),$$

| | where the σ is set to standard deviation of all distances between $x_i^t \in L_t$ and $x_j^t \in U_t$
12 | $\forall x_i^t \in L_t$, compute scores: $\alpha_i = \sum_{j=1}^{n_u} \rho_{ij}$
13 | Sample $\{x_i^t, y_i^t\} \in L_t$ with α_i among top ς into L_t^*
14 **end if**
15 Perform classification on U_t: $[\hat{Y}_t] = f(U_t, L_t^*)$
16 Store all the labeled samples for the next time step:

$$PL_t = \{U_t, \hat{Y}_t\}$$

17 **return** PL_t

difference between the Affinity-based COMPOSE and other variations of COMPOSE is the sampling procedure (steps 4-14 in Algorithm 4). α-Shape and GMM variants instantly extract the core supports based on the seen samples. On the other hand, FAST COMPOSE makes use of all the labeled samples and does not seek for the core-supports.

In contrast to a regular EVLC that updates the classification model regardless of the next batch of data, Affinity-based COMPOSE makes use of a prospective sampling strategy, which means that it samples the labeled data upon the emergence of the upcoming batch. This ties the sampling procedure to the next set of samples, and, thus, those labeled samples with high similarity to the upcoming samples are selected to update the hypothesis space. The algorithm iteratively receives unlabeled samples (step 1 in Algorithm 4) and, in the initial time step, a few labeled samples for the sake of semi-supervised learning (steps 2-4 in Algorithm 4). From then on, if an external update becomes available, the priority will be given to the external update, these labeled samples are directly used to construct a model (steps 5-8 in Algorithm 4). This is due to the importance of the novel information provided by the user. Otherwise, the algorithm uses the internal update that was generated in the previous time step (steps 8-10 in Algorithm 4). To perform semi-supervised learning, an affinity matrix is initially formed between the labeled and the unlabeled samples (step 11 in Algorithm 4). Then, a score is computed based on the sum of the pairwise similarities for each of the labeled samples (step 12 in Algorithm 4). Those samples with score values among the top r percent are then selected to construct a classification model (steps 13-15 in Algorithm 4).

Such a procedure allows to better follow the drift direction in the data stream, when the drift is more intense. It then adapts itself more effectively to the changes. Furthermore, the gap between the drifting samples and the extracted core-supports in the two initial variants of COMPOSE (α-Shape and GMM) gradually increases. However, this issue is avoided in Affinity-based and FAST COMPOSE.

Both FAST COMPOSE and Affinity-based COMPOSE are very fast in regards to their runtimes; however, since the FAST COMPOSE uses all the available pseudo-labeled samples, any outlier and noise can also be included in the constructed hypothesis. Furthermore, addition of the sampling phase into Affinity-based COMPOSE makes it more robust and helps to overcome these issues to a large extent.

5.3 Experimental Results

To assess the performance of the proposed DISCOVERY framework, four different experiments are conducted.

5.3.1 Experimental Setting

All the signals for the experiments are obtained from the CWRU data center [21]. The data contains various defects such as Inner, Outer, and Ball damage under different conditions such as different speeds (1730, 1750, 1772, 1797rpm) and defect widths (0.007, 0.014, 0.021in). The utilized test rig consists of a 2hp Reliance Electric Motor, and the faults were seeded though the drive end of the electric motor. The signal acquisition is performed at a 12kHz sampling rate. The vibrational signals are normalized and, then, divided into non-overlapping segments. Then, a pool of statistical features is extracted from each segment. This pool is further reduced in size through a wrapper feature selection procedure [8]. According to the rank attained through the feature selection, root mean square (RMS) and entropy are the most discriminant features for the designed experiments, respectively. The best two features are used in this study for the following reasons: (i) different numbers of features have been evaluated by means of the wrapper feature selection, and the experiments show that increasing the number of features does not significantly improve the classification accuracy; (ii) while increasing the number of features poses more computational cost and significantly increases the run time at each time step; (iii) α-Shape, which is

Table 5.1 – The characteristics of each experiment including IM speed (rpm) and the defect widths (in) at each time step t.

Experiments	Condition	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
Experiment 1	Speed	1797	1772	1730	1750	1772	1797	1730	1750	1730	1750
	Outer	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007
	Ball	0.007	0.007	0.007	0.007	0.014	0.014	0.021	0.021	0.021	0.021
	Inner	N/A	N/A	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007
Experiment 2	Speed	1750	1772	1730	1750	1730	1797	1772	1750	1797	1730
	Outer	0.007	0.007	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
	Ball	N/A	N/A	0.014	0.014	0.014	0.021	0.021	0.021	0.021	0.021
	Inner	0.007	0.007	0.007	0.021	0.021	0.021	0.021	0.021	0.021	0.021
Experiment 3	Speed	1772	1730	1797	1750	1797	1772	1772	1750	1730	1797
	Outer	0.007	0.007	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
	Ball	0.007	0.007	0.007	0.007	0.014	0.014	0.014	0.021	0.021	0.021
	Inner	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.021
Experiment 4	Speed	1730	1797	1772	1797	1750	1730	1750	1772	1772	1797
	Outer	0.007	0.007	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
	Ball	0.007	0.007	0.007	0.007	0.014	0.014	0.014	0.021	0.021	0.021
	Inner	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.021

the key component for this comparative study, has the maximum efficiency with two features [14, 72, 73]; (iv) entropy shows the highest sensitivity to the emergence of a new class of fault, and, thus, reduces the detection delay.

Each experiment contains ten time steps, and each time step includes a batch of 120 samples per class of fault, where each sample contains two statistical features (i.e., entropy and root mean square). Drift is simulated at each time step by changing the motor speed, the load and the defect widths as reported in Table 5.1. The first two experiments are designed to assess the performance in the presence of an unknown defect. In the next two experiments, concept drift $\mathcal{CD}_A$ takes place without a novel defect. In this study, a unique class label was assigned to all samples representative of a specific type of defect (e.g., Outer, Ball, and Inner) with all possible widths. However, in the initial step $t = 1$, only representative samples of the smallest defect width (partially labeled set of samples) are fed into the diagnostic system.

5.3.2 Two-stage detector

Fig. 5.2(a,b) show the results of the E-CUSUM detector (steps 1-12 in Algorithm 3) on both dimensions of the second experiment. Fig. 5.2(c) shows the result of the $\mathcal{NC}$ detector in the second experiment, where a new class has been detected at $t = 3$. Upon presence of a new unknown defect in the system, a $\mathcal{CD}_A$ at $t = 3$ is detected after processing 251-th sample of Q by means of the first feature, which is the 11-th sample of U_3. The second feature, however, is less sensitive to the presence of an abrupt change in this experiment, and, thus, $\mathcal{CD}_{\mathcal{A}}$ is only detected upon processing 177-th sample of U_3. Nevertheless, this delay does not affect the diagnostic performance, which is due to the use of a parallel detection setting, which detects, $\mathcal{CD}_{\mathcal{A}}$ on the first feature and activates an alarm. Note that the proposed framework receives batches of data subsequently, and all the data in U_t is processed at t. The aim of this work is to design a fast and accurate framework. As a result, at this stage, the exact location of the detected drift in U_t is not a major concern. Moreover, a drift may be gradual w.r.t. one feature, and abrupt in another one. Thus, it is very usual for the detector to detect $\mathcal{CD}_{\mathcal{A}}$ based on one feature.

After a $\mathcal{CD}_{\mathcal{A}}$ is detected, further analysis is performed (steps 15-25 in Algorithm 3) by the $\mathcal{NC}$ detector to determine the addition of unknown new classes to the data stream (steps 26-28 in Algorithm 3).

5.3.3 EVLC module

In the EVLC module, SCARGC and variants of COMPOSE are used along with Affinity-based COMPOSE for the sake of diagnostic bearing defects in IMs. The diagnostic results are compared and ranked in terms of stability and accuracy, as shown in Table 5.2. Each experiment is repeated 20 times. Table 5.2 represents the average and standard deviation of the classification accuracies over 20 runs for each technique. The most right column of Table 5.2 ranks different EVLC algorithms in

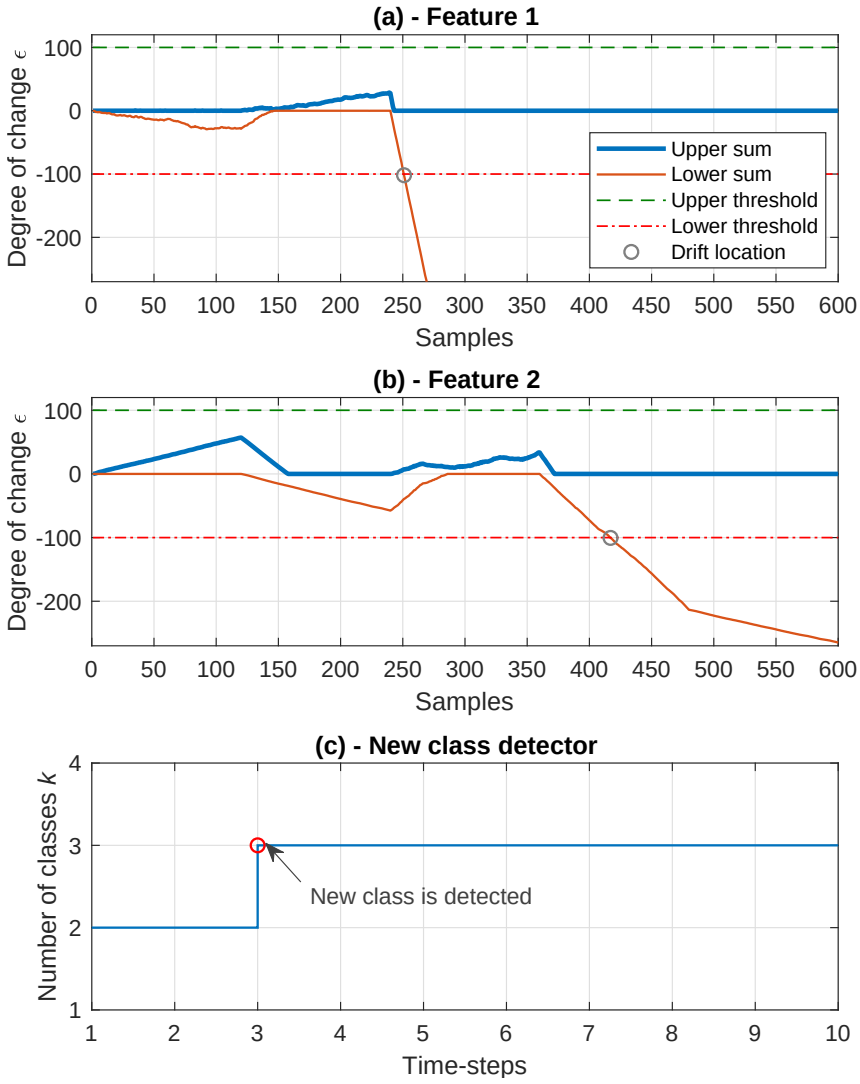

Figure 5.2 – The detection of an abrupt drift in the second experiment. (a) and (b) show the results of the E-CUSUM detector for the first and the second features respectively, in the third time step, $t = 3$. E-CUSUM merges U_2 (240 samples of two classes) and U_3 (360 samples of three classes), resulting into 600 samples (x-axis in the first two panels). (c) The detection of the third class by $\mathcal{NC}$ detector at $t = 3$.

terms of accuracy.

Stability

Each panel of Fig. 5.3 presents the distribution of the classification performances attained by each EVLC in each experiment. Each box in a panel contains all the

Table 5.2 – Average accuracies and standard deviation over all time steps and runs attained by each technique.

Experiments	Algorithms	Accuracy	Std.	Rank
Experiment 1	FAST COMPOSE	95.24	11.27	4
	Affinity-based COMPOSE	98.76	5.60	2
	COMPOSE (α-Shape)	**99.75**	**0.83**	1
	COMPOSE (GMM)	97.93	7.62	3
	SCARGC	66.66	25.81	5
Experiment 2	FAST COMPOSE	81.66	11.64	4
	Affinity-based COMPOSE	83.28	11.21	3
	COMPOSE (α-Shape)	84.99	9.79	2
	COMPOSE (GMM)	**86.99**	**8.02**	1
	SCARGC	36.3	26.07	5
Experiment 3	FAST COMPOSE	93.32	13.25	4
	Affinity-based COMPOSE	**96.58**	**6.32**	1
	COMPOSE (α-Shape)	87.29	16.16	5
	COMPOSE (GMM)	95.55	9.78	2
	SCARGC	94.54	8.29	3
Experiment 4	FAST COMPOSE	93.70	12.99	4
	Affinity-based COMPOSE	**95.20**	**9.56**	1
	COMPOSE (α-Shape)	87.26	17.65	5
	COMPOSE (GMM)	93.71	12.60	3
	SCARGC	93.86	9.90	2

attained accuracies (solid circles) by each EVLC (10 time steps $\times$ 20 runs). Fig. 5.3 shows that, in the first two experiments, α-Shape, GMM, and Affinity-based COMPOSE are stable, where they, with a slight difference, are ranked from first to third, respectively, based on the calculated standard deviations as listed in Table 5.2. FAST COMPOSE, however, is not stable. This is due to the fact that the algorithm uses all the classified samples for the next time step. This results in the lower classification quality, and, then, the higher standard deviation. SCARGC, on the other hand, is the most unstable technique in the first set of experiments. Such an instability is the result of the structure of the SCARGC algorithm, where it assumes that the number of classes are fixed, and as a result, upon occurrence of a new class of fault, it is doomed to misclassify samples of the new class.

In the last two experiments, where the number of classes remains unchanged, the

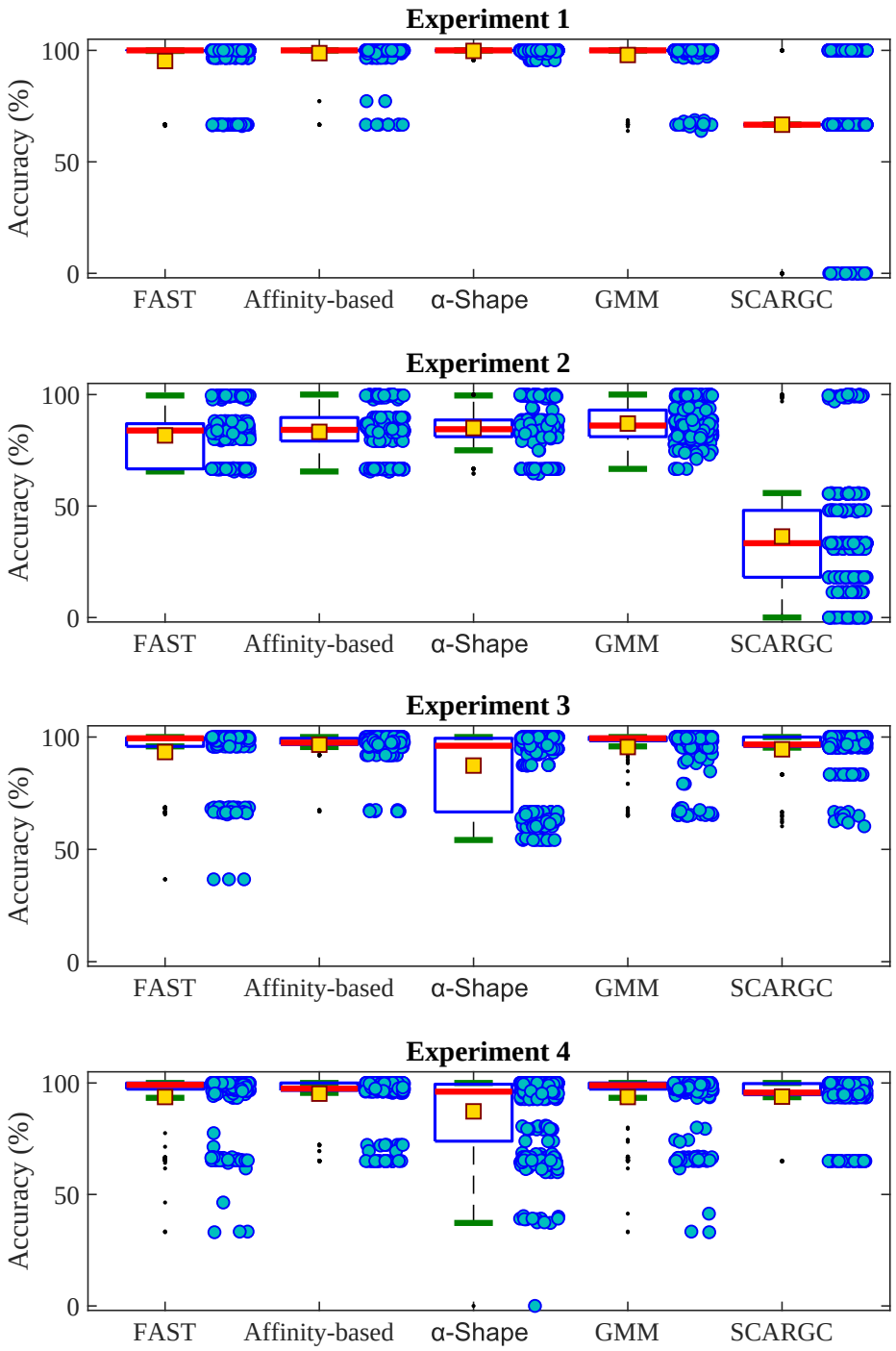

Figure 5.3 – The distribution of the accuracies (solid circles) attained by each EVLC in each designed experiment. The mean values are specified by solid squares, and outliers are shown by dotes.

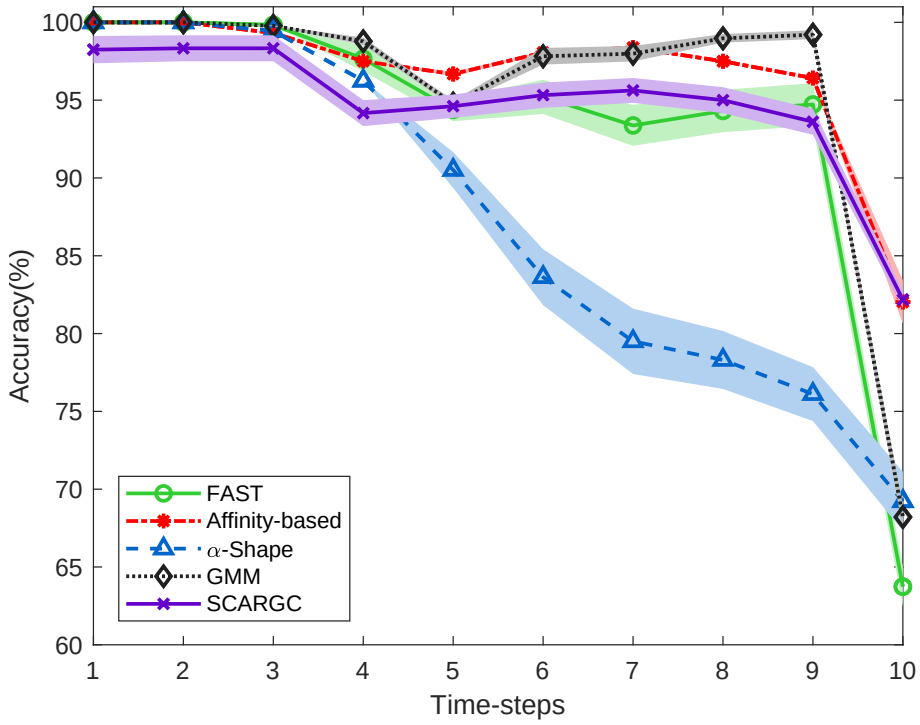

Figure 5.4 – The accuracy profile of each EVLC technique for all 20 runs of the third experiment. The standard errors are shown by the shaded areas around the mean curves.

Affinity-based COMPOSE is the most stable algorithm. The second and third ranks are assigned to SCARGC and GMM, respectively. Here, the good performance of SCARGC is due to the nature of the experiments, in which the number of classes is fixed in the data stream. Same as the previous two experiments, FAST COMPOSE is unstable to some extent due to the aforementioned issue. However, α-Shape turns out to be the most unstable algorithm in the last two experiments. This is due to the fact that the pace of drift in the last two experiments is faster and harsher than the first two experiments, which increases the gap between the drifted samples and the extracted core-supports. Nevertheless, the stability of GMM remains unchanged in the last two experiments, since the data stream follows a Gaussian distribution.

Accuracy

In the first two experiments, GMM is followed by α-Shape, and Affinity-based COMPOSE as second and third ranks, with a slight difference, respectively. FAST COMPOSE and SCARGC are ranked third and fifth in terms of accuracy (see Table 5.2). In the last two experiments, the Affinity-based COMPOSE outperforms the other competitors in terms of accuracy. This can be seen in Table 5.2, and moreover for the third experiment, in Fig. 5.4. It is followed by GMM, SCARGC, FAST, and α-Shape COMPOSE as subsequent ranks, respectively. These rankings are based on averaging the accuracies for each experiment over 20 runs, as reported in Table 5.2. Fig. 5.4 shows the standard errors at each time step in shaded areas around the averaged accuracies over 20 runs in the third experiment.

5.3.4 Discussion

This section studies the efficiency and efficacy of the EVLC algorithms for diagnosing bearing defects in IMs. SCARGC assumes the number of classes in the data stream remains unchanged, which limits its usage in the proposed framework. Even though, COMPOSE implementations are quite flexible in this sense, still, each of them have their own limitations. The employed Delauny tessellation in α-Shape complexity is $O(n^{\lfloor(\gamma+1)/2\rfloor})$, where n and γ are the number of samples and dimensions, that is, as the γ increases, the efficiency falls drastically. GMM is not limited in this sense; however, it always assumes that the data follows a Gaussian distribution. FAST COMPOSE sacrifices accuracy and stability to some extent, which makes it the fastest COMPOSE. The proposed Affinity-based COMPOSE increases the performance of FAST COMPOSE by resorting to a prospective sampling strategy. Aside from its efficiency, the results show that it is always among the top ranks.

The real-time property of the designed framework is analyzed in terms of runtime. Table 5.3 contains the average runtimes recorded for each technique over different

Table 5.3 – The averaged run-time of the DISCOVERY diagnostic framework over all scenarios and runs.

Algorithms	Time (s)	Rank
FAST COMPOSE	0.0027	1
Affinity-based COMPOSE	0.0032	2
COMPOSE (α-Shape)	0.8533	5
COMPOSE (GMM)	0.4151	3
SCARGC	0.6591	4

runs and scenarios. FAST COMPOSE is the fastest technique, and with a slight difference, Affinity-based COMPOSE stands in the second rank. GMM, SCARGC, and α-Shape, on the other hand, are slower and ranked from third to fifth, respectively. The recorded runtimes show that the framework has an acceptable speed with all EVLC algorithms. Affinity-based COMPOSE is then the most efficient algorithm considering both accuracy and speed into account. Thus, the proposed framework is highly practical to be used for real-time applications. All tests have been performed in Matlab R2017b on a computer with Intel Core i7 6700HQ CPU and 8GB of RAM.

5.4 Summery

In this chapter, we proposed a novel framework to diagnose faults in non-stationary environments. The proposed framework is able to dynamically update the diagnostic model and adapt it with respect to the changes in the evolving data stream. A two-stage detector has been presented in the proposed framework to detect abrupt and gradual concept drifts including presence of new class of defects in the data stream. Extended Cumulative SUM (E-CUSUM) detects abrupt changes, and the novelty detector determines the presence of new unseen classes. To track the gradual concept drift in the data stream, which is partially labeled in the initial time step; the proposed framework makes use of an EVLC module for fault classification.

Various EVLC algorithms are integrated with the proposed framework and their

performances are compared in diagnosing bearing defects in IMs. This work also proposes a novel EVLC algorithm, called Affinity-based COMPOSE, to improve the performance. The attained results confirm that the Affinity-based COMPOSE outperforms other competitors in terms of both accuracy and speed. Although the diagnostic framework is only provided with prior information about possible faults at the initial step; however, the attained results show that it still detects novel faults without receiving any update.

Chapter 6

Conclusion and Remarks

The main goal of this work is to design an efficient yet accurate data-driven diagnostic with the limited supervision, that is only a few numbers of labeled samples along with a large number of unlabeled samples are available. This has been accomplished by restoring to semi-supervised learning approaches, which bring about two main advantages to the system. Firstly, it eliminates the need for expensive human labor that is required to prepare a large number labeled samples as a training set. In addition, in contrast to the widely used approaches, for unsupervised and supervised learning, semi-supervised learning exploits both available labeled and unlabeled samples resulting in a more precise diagnostic model.

Four different diagnostic frameworks are proposed in this thesis to address the limited supervision problem. These frameworks generally contain three main modules. The first module is for extracting informative features from the input signals. This is while there is no guarantee that all of the generated features are useful as there may exist some non-informative or similar features, which result in overfitting and performance-drop. Thus, the second module is considered for selecting the best features or mapping them to a different feature space to attain a smaller set of informative features, which is later fed to the third module for decision making. In order to attain the best quality features, only semi-supervised feature reduction methods are considered. Through the design of the first diagnostic framework, a comparative study has been performed that is focused on the impact of semi-supervised dimensionality reduction techniques. Based on this study, SDA seems to be a reliable choice for fault diagnosis in IMs.

The second diagnostic framework, on the other hand, is a hybrid scheme that has more feature extraction techniques and also enables a comparative study of the semi-supervised classification algorithms. Various state-of-the-art classifiers are employed and compared in this framework. Moreover, a novel semi-supervised classifier, called S3AL, has been proposed to maximize the performance of this framework. S3AL makes use of the concept of surface estimation to capture the structure of the data to enable an SSL procedure. The results show that S3AL has a superior performance and is a reliable choice for diagnosing bearing defects in IMs..

The third diagnostic framework is proposed to address a more challenging problem. While feature reduction techniques can be very helpful in preparing informative sets of features for decision making, it is possible that the results are not satisfying as expected. This may occur due to issues such as incompatibility of the feature reduction and classification algorithms, or the presence of an extremely high-dimensional feature space. Thus, the third diagnostic framework utilizes a semi-supervised deep learning procedure to address the problem of fault diagnosis with the limited supervision in the high-dimensional feature space. In addition, various state-of-the-art classifiers and deep learning algorithms are compared in this study. The attained results show that Semi-supervised Deep Ladder Network has a superior performance both in terms of accuracy and convergence rate.

The last framework is designed to address the most critical problem in this work. Although the first three frameworks and most of the proposed diagnostic schemes in the industry are designed for online diagnostic procedures, their diagnostic model is trained in an offline manner. This may not cause any problem in many cases; however, there are non-stationary conditions in the industry in which the input space is, in fact, a data stream and it may evolve and change in time. In such a condition, a dynamic diagnostic model is required since an offline model cannot interpret new conditions. The proposed framework, called DISCOVERY, exploits semi-supervised classifiers that are able to operate in a non-stationary environment. In order to maximize the

speed and efficiency of the diagnostic framework, a novel semi-supervised classifier has been proposed that is named Affinity-based COMPOSE. The results show that Affinity-based COMPOSE has superior performance compared to its rivals. However, such algorithms assume that only gradual drifts may take place in the data stream, albeit not always being the case. Thus, a double-stage detector module is designed and used in DISCOVERY, which is able to detect various types of drifts including the presence of the new class in the system. DISCOVERY then generates internal updates and adapts w.r.t. abrupt changes. The results indicate that DISCOVERY is reliable and able to address the problem of fault diagnosis under the limited supervision in non-stationary environments.

References

[1] R. Razavi-Far, E. Hallaji, M. Saif, and L. Rueda, "A hybrid scheme for fault diagnosis with partially labeled sets of observations," in *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 61–67.

[2] R. Razavi-Far, E. Hallaji, M. Saif, and G. Ditzler, "A novelty detector and extreme verification latency model for nonstationary environments," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2018.

[3] J. Yi, D. Huang, H. He, W. Zhou, Q. Han, and T. Li, "A novel framework for fault diagnosis using kernel partial least squares based on an optimal preference matrix," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4315–4324, 2017.

[4] M. Kang, J. Kim, L. M. Wills, and J. M. Kim, "Time-varying and multiresolution envelope analysis and discriminative feature analysis for bearing fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7749–7761, 2015.

[5] K. A. Loparo, M. L. Adams, W. Lin, M. F. Abdel-Magied, and N. Afshari, "Fault detection and diagnosis of rotating machinery," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1005–1014, 2000.

[6] G. Georgoulas, V. Climente-Alarcon, J. A. Antonino-Daviu, I. P. Tsoumas, C. D. Stylios, A. Arkkio, and G. Nikolakopoulos, "The use of a multilabel classification framework for the detection of broken bars and mixed eccentricity faults based on the start-up transient," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 625–634, 2017.

[7] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2012.

[8] T. W. Rauber, F. de Assis Boldt, and F. M. Varejo, "Heterogeneous feature models and feature selection applied to bearing fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 637–646, 2015.

[9] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.

[10] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2016.

[11] R. Razavi-Far and M. Saif, "Ensemble of extreme learning machines for diagnosing bearing defects in non-stationary environments under class imbalance condition," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–6.

[12] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Adaptive strategies for learning in nonstationary environments: a survey," *Computational Intell. Mag.*, vol. 10, no. 4, pp. 12–25, 2015.

[13] M. Teichmann and M. Capps, "Surface reconstruction with anisotropic density-scaled alpha shapes," in *Visualization '98. Proc.*, 1998, pp. 67–72.

[14] K. B. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 12–26, Jan 2014.

[15] M. Kang, M. R. Islam, J. Kim, J. M. Kim, and M. Pecht, "A hybrid feature selection scheme for reducing diagnostic performance deterioration caused by outliers in data-driven diagnostics," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3299–3310, May 2016.

[16] R. Razavi-Far, M. Farajzadeh-Zanjani, and M. Saif, "An integrated class-imbalance learning scheme for diagnosing bearing defects in induction motors," *IEEE Transactions on Industrial Informatics*, 2017.

[17] M. Farajzadeh-Zanjani, R. Razavi-Far, M. Saif, J. Zarei, and V. Palade, "Diagnosis of bearing defects in induction motors by fuzzy-neighborhood density-based clustering," in *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 935–940.

[18] R. Razavi-Far and M. Kinnaert, "A multiple observers and dynamic weighting ensembles scheme for diagnosing new class faults in wind turbines," *Control Eng. Practice*, vol. 21, no. 9, pp. 1165–1177, 2013.

[19] R. Razavi-Far, V. Palade, and E. Zio, "Optimal detection of new classes of faults by an invasive weed optimization method," in *International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 91–98.

[20] J. Zarei, M. A. Tajeddini, and H. R. Karimi, "Vibration analysis for bearing fault detection and classification using an intelligent filter," *Mechatronics*, vol. 24, no. 2, pp. 151–157, 2014.

[21] "Case western reserve university bearing data center website." [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

[22] R. B. Randall, *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.

[23] W. Q. Wang, F. Ismail, and M. F. Golnaraghi, "Assessment of gear damage monitoring techniques using vibration measurements," *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 905–922, 2001.

[24] T. Loutas, G. Sotiriades, I. Kalaitzoglou, and V. Kostopoulos, "Condition monitoring of a single-stage gearbox with artificially induced gear cracks utilizing online vibration and acoustic emission measurements," *Applied Acoustics*, vol. 70, no. 9, pp. 1148–1159, 2009.

[25] M. Delgado-Prieto and D. Zurita-Millan, "Chromatic monitoring of gear mechanical degradation based on acoustic emission," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8707–8717, 2017.

[26] M. H. Marzebali, J. Faiz, G. A. Capolino, S. H. Kia, and H. Henao, "Gear fault detection based on stator current and mechanical torque signatures of wound rotor induction generator," *IEEE Transactions on Energy Conversion*, vol. PP, no. 99, pp. 1–1, 2018.

[27] S. H. Kia, H. Henao, and G. A. Capolino, "Gear tooth surface damage fault detection using induction machine stator current space vector analysis," *IEEE Transactions on Industrial Electronics*, vol. 62(3), pp. 1866–1878, 2015.

[28] ——, "Fault index statistical study for gear fault detection using stator current space vector analysis," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 4781–4788, Nov 2016.

[29] J. R. Ottewill and M. Orkisz, "Condition monitoring of gearboxes using synchronously averaged electric motor signals," *Mechanical Systems and Signal Processing*, vol. 38, no. 2, pp. 482–498, 2013.

[30] J. Ottewill, A. Ruszczyk, and D. Broda, "Monitoring tooth profile faults in epicyclic gearboxes using synchronously averaged motor currents: Mathematical

modeling and experimental validation," *Mechanical Systems and Signal Processing*, vol. 84, pp. 78–99, 2017.

[31] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[32] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," *Optimization Methods and Software*, vol. 15, pp. 29–44, 2001.

[33] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 129–143, 2011.

[34] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–7.

[35] D. Zhang and Z. hua Zhou Songcan Chen, "Semi-supervised dimensionality reduction," in *In: Proceedings of the 7th SIAM International Conference on Data Mining*, 2007, pp. 11–393.

[36] F. Nie, D. Xu, I. W. H. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1921–1932, July 2010.

[37] Y. Liu, F. Nie, J. Wu, and L. Chen, "Efficient semi-supervised feature selection with noise insensitive trace ratio criterion," *Neurocomputing*, vol. 105, pp. 12–18, 2013.

[38] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14(6), pp. 1662–1672, 2012.

[39] Y. Bengio, O. Delalleau, and N. Le Roux, *Label Propagation and Quadratic Criterion*. MIT Press, 2006, p. 193216.

[40] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th Int. Conf. Machine Learning*, 2001, pp. 19–26.

[41] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *AISTATS*, 2005, pp. 57–64.

[42] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, Dec 2014.

[43] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Machine Learning*, 1999, pp. 200–209.

[44] K. P. Bennett, A. Demiriz, and R. Maclin, "Exploiting unlabeled data in ensemble methods," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2002, pp. 289–296.

[45] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31(11), pp. 2000–2014, 2009.

[46] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *IN ICML*, 2003, pp. 912–919.

[47] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[48] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9(4), pp. 2226–2238, 2013.

[49] Z. Ge, S. Zhong, and Y. Zhang, "Semisupervised kernel learning for fda model and its application for fault classification in industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 12(4), pp. 1403–1411, 2016.

[50] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A highly-accurate framework for self-labeled semi-supervised classification in industrial applications," *IEEE Transactions on Industrial Informatics*, vol. PP(99), pp. 1–1, 2017.

[51] M. Grbovic, W. Li, N. A. Subrahmanya, A. K. Usadi, and S. Vucetic, "Cold start approach for data-driven fault detection," *IEEE Transactions on Industrial Informatics*, vol. 9(4), pp. 2264–2273, 2013.

[52] W. Jiang, Z. Zhang, F. Li, L. Zhang, M. Zhao, and X. Jin, "Joint label consistent dictionary learning and adaptive label prediction for semisupervised machine fault classification," *IEEE Transactions on Industrial Informatics*, vol. 12(1), pp. 248–256, 2016.

[53] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[54] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22(4), pp. 469–483, 1996.

[55] H. Keskes and A. Braham, "Recursive undecimated wavelet packet transform and DAG SVM for induction motor diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1059–1066, 2015.

[56] Y. Li, M. Xu, X. Liang, and W. Huang, "Application of bandwidth EMD and adaptive multi-scale morphology analysis for incipient fault diagnosis of rolling bearings," *IEEE Transactions on Industrial Electronics*, vol. PP, no. 99, pp. 1–1, 2017.

[57] Y. Tian, J. Ma, C. Lu, and Z. Wang, "Rolling bearing fault diagnosis under variable conditions using LMD-SVD and extreme learning machine," *Mechanism and Machine Theory*, vol. 90, pp. 175–186, 2015.

[58] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-supervised learning with ladder networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, ser. NIPS, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 3546–3554.

[59] H. Jiang, J. Chen, G. Dong, T. Liu, and G. Chen, "Study on hankel matrix-based SVD and its application in rolling element bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 52-53, pp. 338–359, 2015.

[60] H. C. Hsueh and S. Y. Chien, "On-line local mean decomposition and its application to ecg signal denoising," in *IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, Oct 2014, pp. 17–20.

[61] Y. Wang, Z. He, and Y. Zi, "A comparative study on the local mean decomposition and empirical mode decomposition and their applications to rotating machinery health diagnosis," *Journal of Vibration and Acoustics*, vol. 132, pp. 021 010–021 010–10, 2010.

[62] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio, "Deconstructing the ladder network architecture," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. PMLR, 2016, pp. 2368–2376.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[64] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, Nov 2001.

[65] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010.

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations*, 2015.

[67] G. Ditzler and R. Polikar, "Semi-supervised learning in nonstationary environments," in *Int. Joint Conf. Neural Netw.*, 2011, pp. 2471–2478.

[68] J. Kolter and M. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *J. Machine Learning Res.*, vol. 8, pp. 2755–2790, 2007.

[69] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, 2011.

[70] G. Krempl, *The Algorithm APT to Classify in Concurrence of Latency and Drift*. Springer Berlin Heidelberg, 2011, pp. 222–233.

[71] V. M. A. Souza, D. F. Silva, J. Gama, and G. E. A. P. A. Batista, *Data Stream Classification Guided by Clustering on Nonstationary Environments and Extreme Verification Latency*, pp. 873–881. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611974010.98

[72] R. Capo, A. Sanchez, and R. Polikar, "Core support extraction for learning from initially labeled nonstationary environments using compose," in *2014 International Joint Conference on Neural Networks (IJCNN)*, July 2014, pp. 602–608.

[73] M. Umer, C. Frederickson, and R. Polikar, "Learning under extreme verification latency quickly: Fast compose," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–8.

[74] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers; part i: Detecting nonstationary changes," *IEEE Trans. Neural Netw.*, vol. 19, pp. 1145–1153, 2008.

[75] M. Yesilbudak, "Clustering analysis of multidimensional wind speed data using k-means approach," in *IEEE Int. Conf. Renewable Energy Res. and Appl. (ICRERA)*, 2016, pp. 961–965.

Appendix A

Copyright Permissions

Title: A Hybrid Scheme for Fault Diagnosis with Partially Labeled Sets of Observations

Conference Proceedings: Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on

Author: Roozbeh Razavi-Far

Publisher: IEEE

Date: Dec 2017

Copyright © 2017, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.

Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Vita Auctoris

Ehsan Hallaji was born in 1992 in Tehran, Iran. He pursued his undergraduate studies in the Department of Computer Engineering at the Shahid Rajaee University, Iran, and received B.Sc. degree in software engineering in 2015. He is currently a candidate for the M.A.Sc. degree in Electrical and Computer Engineering at the University of Windsor, Canada and expects to graduate in Spring 2018. He has multidisciplinary background in electrical engineering, software engineering, computer networks and computer science. His research area mainly involves machine learning, computational intelligence, artificial intelligence, data mining, and their application in the design of diagnostic systems.